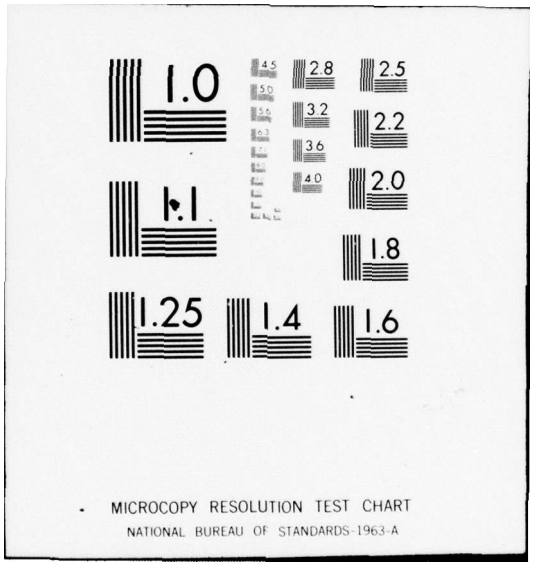
AD-A048 181 TEXAS A AND M UNIV COLLEGE STATION DEPT OF ELECTRICA--ETC F/G 17/4
LOW COST ANTI-JAM DIGITAL DATA-LINKS TECHNIQUES INVESTIGATIONS.(U)
JUN 77 J H PAINTER, J N HOLYOAK, C J YOON F33615-75-C-1011
UNCLASSIFIED AFAL-TR-77-104 NL

1 of 3
ADA048 181


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A048181

AFAL-TR-77-104

LOW COST ANTI-JAM DIGITAL DATA-LINKS TECHNIQUES INVESTIGATIONS

Department of Electrical Engineering
Texas A&M University
College Station, Texas 77843

June 1977

Technical Report AFAL-TR-77-104

Final Report for the Period 1 July 1974 - 30 September 1976

Approved for Public Release; Distribution Unlimited.

AIR FORCE AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

John W Mayhan

JOHN W. MAYHAN
Project Engineer

CHARLES C. GAUDER
Chief, System Development Branch
System Avionics Division

FOR THE COMMANDER

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFAL/AAD, W-PAFB, OH 45433 to help us maintain a current mailing list".

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFAL-TR-77-104	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* LOW COST ANTI-JAM DIGITAL DATA-LINKS TECHNIQUES INVESTIGATIONS, FINAL REPORT FOR PHASE I		5. TYPE OF REPORT & PERIOD COVERED FINAL REPORT 1 Jul 74 - 30 Sep 76 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Dr. John H. Painter Mr. Joel N. Holyoak Mr. Chang J. Yoon		8. CONTRACT OR GRANT NUMBER(s) F33615-75-C-1011
9. PERFORMING ORGANIZATION NAME AND ADDRESS Texas A&M University Department of Electrical Engineering College Station, Texas		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project No. 7662 Task No. 766204 Work Unit No. 76620428
11. CONTROLLING OFFICE NAME AND ADDRESS System Avionics Division (AA) Air Force Avionics Laboratory (AFSC) Wright-Patterson Air Force Base, Ohio		12. REPORT DATE 1 June 77 13. NUMBER OF PAGES 217
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for Public Release; Distribution Unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Signal Processing
Interference Cancelation
Optimum Demodulation
Anti-Multipath Receiver
Anti-Jam Receiver
Recursive Maximum Likelihood Demodulation
Integrated Detection, Estimation and Identification (IDEI)

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This is a final report on the first phase of an investigation into new techniques for communicating digital data between aircraft and other terminals. The impetus behind the research is the necessity to develop low-cost, high performance digital data-links for a variety of aircraft operating in a variety of interference environments. The goal of the investigation is to develop real-time, sampled-data processing techniques to combat both natural and intentional interferences on aeronautical radio navigation and data-link channels. Such

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

BLOCK 20 (contd):
processing techniques, if not overly complex, offer the potential for low-cost implementation. In order not to make a Priori assumptions which would force high cost implementation, a processing approach is sought which is relatively independent of signal modulation type. Thus, the thrust is toward signal processors which operate essentially as modems on the output of standardized, unsophisticated radio receivers. A certain amount of the investigative effort concerns mathematical modeling of the multipath perturbations and of possible additive interfering signals. Advantage is taken of prior NASA work in anti-multipath reception. In the present effort, the restricted NASA multipath channel model is generalized sufficiently to be realistic for narrow or wide-band signal modulations. Next, the detection of digital signals in the postulated channel is considered in the most fundamental framework of probabilistic decision theory. Finally, a Monte Carlo simulation program is designed and tested, for exercising the detection algorithms. Primarily, the present investigation examines whether the same processing approach which was successful for multipath channels will also be successful in the presence of intentional additive colored interference.

PREFACE

The work covered by this report was accomplished by the Department of Electrical Engineering, Texas A&M University, College Station, Texas, under Air Force Contract F33615-75-C-1011. The effort was documented under Project 7662, Avionic Data Transmission and Reception, Task No. 766204, Anti-Jam Processing Techniques for Multifunction Application, Work Unit No. 76620424, Low Cost Anti-Jam Digital Data-Links Techniques Investigations. The contract monitor was John W. Mayhan (AFAL/AAD) of the Air Force Avionics Laboratory, Wright-Patterson Air Force Base, Ohio.

From 1971 through 1973, the Principal Investigator developed a new sampled-data processing technique for anti-multipath reception of aeronautical data-link signals. The research, which was performed at NASA Langley Research Center, included development of a Monte Carlo computer simulation for testing the processing algorithms, as well as performance of a flight experiment to verify the multipath mathematical model. The multipath perturbations, for a diffuse Doppler-spread channel, took the form of complex multiplicative noise in the complex signal domain.

The anti-multipath research yielded a new type of detection algorithm which works by tracking and cancelling the multiplicative noise. A patent was eventually obtained on the new receiver.[1] After some experience was gained through simulation of the anti-multipath detector, it became obvious to the Principal Investigator that the same basic processing approach would probably work for channels characterized by colored additive interference. Such channels occur in the presence of Radio Frequency Interference or Jamming.

In 1974, a proposal was made to the Air Force Avionics Laboratory for the research presently being reported. The scope of the proposed effort was essentially to (i) extend the NASA multipath model; (ii) derive detection algorithms for multipath and additive colored interference; and (iii) produce an upgraded Monte Carlo simulation program for testing the new algorithms.

---

[1] John H. Painter, "Anti-multipath Digital Signal Detector," United States Patent # 3,984,634, October 5, 1976.

PREFACE (Continued)

The present report is on the outcome of the effort from July 1974 through October 1976. In short, the research was successful. New anti-multipath, anti-jamming algorithms have been derived. First simulation tests show that they work well. A follow-on effort has been defined for pursuing the reduction of the detection theory to practice. The follow-on effort has been funded by AFAL for a period ending in December 1977. The present report forms the baseline for the follow-on effort.

LIST OF RESEARCH PERSONNEL

NAME	TITLE AND TASK
Dr. John H. Painter	Principal Investigator
Mr. Joel N. Holyoak	Research Associate for GFE Computer Facility
Mr. Chang J. Yoon	Research Assistant for (i) Simulation (ii) Identification
Ms. Kathy L. Wright	Secretarial Assistant for Report Preparation

The Department of Electrical Engineering
Texas A&M University
College Station, Texas 77843
(713) 845-7441

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

INTRODUCTION

This is a report on the first phase of an investigation into new techniques for communicating digital data between aircraft and other terminals. The impetus behind the research is the necessity to develop low-cost, high performance digital data-links for a variety of aircraft operating in a variety of interference environments. Currently there exists no standard aeronautical data-link system which combines low-cost and high-performance. Presently, systems which require high (60 db.) protection against intentional interference use signalling techniques which result in high cost receivers. Systems requiring less protection result in lower cost receivers but are not standardized. The lack of a standard system contributes somewhat to higher cost for both high protection and low protection hardware.

The goal of the investigation is to develop real-time, sampled-data processing techniques to combat both natural and intentional interferences on aeronautical radio navigation and data-link channels. Such processing techniques, if not overly complex, offer the potential for low-cost implementation. In order not to make A Priori assumptions which would force high cost implementation, a processing approach is sought which is relatively independent of signal modulation type. Thus, the thrust is toward signal processors which operate essentially as modems on the output of standardized, unsophisticated radio receivers.

Although, the investigation is primarily receiver oriented, some inferences are drawn on the signal design and coding aspects of system optimization. The orientation is also toward recursive (real-time) data processing receivers. Although the analyses and simulations are for sampled-data (discrete-time), the results also have implications for continuous-time systems. The choice of sampled-data analyses and design is because of the immediate implementability (for restricted bandwidths) in digital hardware. Also, the ease of Charge Coupled Device (CCD) implementations for sampled data algorithms influences the choice of sampled-data.

Other influences on the investigation are the desirability to use digital signalling for aircraft which cannot carry directive antennas. The use of

simple antennas insures that the signals are always subject to multipath interference, resulting from Earth surface reflection. To assume directive antennas not only increases cost but also evades the issue of dealing with the interference. Also, additive colored interference, intentional or otherwise, is assumed to be present.

A certain amount of the investigative effort concerns mathematical modeling of the multipath perturbations and of possible additive interfering signals. Advantage is taken of prior NASA work in anti-multipath reception. In the present effort, the restricted NASA multipath channel model is generalized sufficiently to be realistic for narrow or wideband signal modulations. Next, the detection of digital signals in the postulated channel is considered in the most fundamental framework of probabilistic decision theory. Finally, a Monte Carlo simulation program is designed and tested, for exercising the detection algorithms. Primarily, the present investigation examines whether the same processing approach which was successful for multipath channels will also be successful in the presence of intentional additive colored interference.

SECTION II

FORMULATION OF THE INVESTIGATION

2.1. MOTIVATION AND DIRECTION

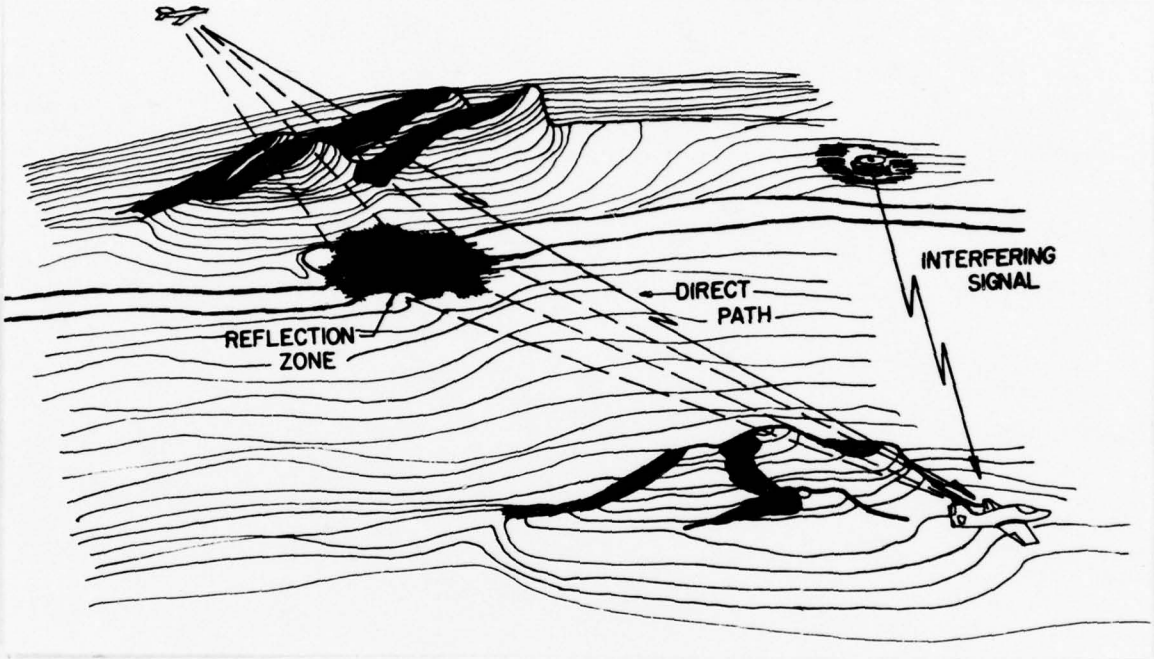


Figure 2.1.-1. The Communication Problem.

This report gives results of a new investigation into the problem of digital communication for aircraft in the presence of ground reflections and other additive interference. Figure 2.1.-1 depicts the type of channels which are dealt with. The aircraft are taken to be relatively unsophisticated, such as Remotely Piloted Vehicles or light tactical aircraft. For reasons of cost and maneuverability, directive antennas are not assumed. Thus, the aircraft are always subject to radio reflections from the Earth's surface. At times during its missions, the aircraft may also be subject to intentional additive interference from other signal sources.

State of the art digital data-links for aircraft which give protection against multipath and intentional additive interference are based on communication theory from the early 1960's. Although current avionics may use new

device technologies, the underlying theories are not new. The new device technologies have increased data processing capabilities many fold over those of the '60's. Thus, it seems timely to re-examine the communication theory to see if the new device technologies can be exploited for aeronautical data-link purposes. The present investigation makes just such a re-examination.

It has long been understood that there are three ways to deal with additive interference in a radio channel. These may be loosely characterized as (i) averaging; (ii) avoidance; and (iii) cancellation. In the case of multipath, signal information may be transmitted simultaneously at two different frequencies and the received signals summed after demodulation. This averages the interference over frequency and is a type of "Diversity" reception [1]. In the case of intentional additive interference, Phase-Shift-Keying may be used for transmission employing a binary Pseudo-Noise (PN) code. Upon reception, the interference is spread in frequency and is essentially averaged. This is a type of "Spread Spectrum" technique [2, 3, 10, 12].

In the PN scheme, the last step in generating the transmitted message signal is to perform a multiplication with a two-level (± 1) PN waveform. The transitions between +1 and -1 levels are generated in a random-appearing, but known sequence. The bandwidth of the PN waveform is much larger than that of the transmitted message signal. Because multiplication in the time domain yields convolution in the frequency domain, the resulting spectrum has essentially the width of the PN waveform.

In detection, the spread signal is multiplied by a replica of the same PN waveform used for generation. This multiplication restores the basic message signal. If the additive interference is due to an undesired signal, the receiver multiplication spreads its spectrum. Because the detector bandwidth need only be as wide as the basic message signal, only a small portion of the additive interference power enters the detector. To the detector, the spread interference looks like additional white noise and is treated accordingly. If the additive interference is due to delayed multipath versions of the desired signal, the PN-reception scheme also rejects the interference to a certain degree, provided the interfering signals lag the desired signal by at least one symbol of the PN sequence.

The chief difficulty with the PN scheme is that of synchronizing the receiver's PN sequence generator to the sequence inherent in the received signal.

In the PN scheme the receiver sequence, which is to multiply the received signal, must be synchronized to the received sequence to within a fraction of a PN symbol. For a 10 Megabit per second PN waveform, synchronization is required to within tens of nano-seconds. Moreover, the received sequence clock rate is not constant, due to changing Doppler conditions. Thus, active synchronization processing is required. The PN synchronization problem is a direct result of spreading the spectrum of the transmitted signal over that which is required to just carry the message information. If the spectrum is spread by six orders of magnitude, then the required synchronization accuracy is increased by six orders of magnitude. It is the synchronization problem for highly spread signals which drives the hardware cost up and out of the low-cost arena.

Interference Avoidance techniques may lead to the diversity method of rejection of the interfering signal through adaptive steering of antenna pattern nulls. This technique can be applied, provided the sources of desired signal and undesired signal, or multipath reflection, are not nearly co-linear, with respect to the receiver. Another avoidance method is that spread-spectrum technique known as "Frequency-Hopping" (FH).

In the frequency-hopping scheme, the basic message signal is shifted in frequency over a collection of spot frequency assignments. The sequence of frequencies appears random but is known. The same kind of random sequence generator is used for FH as was used for PN. In the receiver, the local oscillator is controlled by a replica sequence so that the receiver is instantaneously tuned through the proper set of frequencies. This tuning does not, however, spread the spectrum of an incident additive interfering signal. If the interference is narrow-band, with respect to the width of the set of frequency-hop assignments, then only a few of the transmitted symbols will be affected. Error correcting coding of the basic message signal may be employed to restore the affected symbols. Multipath interference is also avoided by hopping the signal so rapidly that the receiver has left any given frequency before the delayed reflected signal has arrived at that frequency.

Fast FH schemes using many frequency slots encounter synchronization problems similar to those for PN systems. High performance FH systems or hybrid FH/PN systems have not demonstrated low cost to date.

The third method for dealing with interference is that of cancellation. Essentially this means tracking the interference with a filter and, for additive

cases, subtracting the tracked estimate from the received waveform. No documentation or other information has been found during the present investigation which indicates that the Interference Cancellation Strategy has ever been applied to the aeronautical communication problem. There is a good reason for this. Prior to the development of recursive, least-squares estimation (Kalman filter) theory, there was no theoretical basis to support such a cancellation strategy. Thus, interference averaging and avoidance, in the form of spread-spectrum techniques, continue to form the present state-of-the-art in aeronautical data-link and navigation systems [4].

This investigation does not set out to employ cancellation methods, A Priori. What occurs below is that a first principles derivation of optimum recursive digital detection algorithms produces the interference cancelling solution. Thus, interference cancellation is proved to be the optimum scheme under quite general performance criteria. This is perhaps the most interesting result of the entire effort.

It is shown in following sections that interference cancellation requires estimation of the waveform of an additive undesired signal. For multipath interference (in the diffuse case), cancellation requires estimation of the waveform of an equivalent multiplicative interference process. When both types of interference are present, simultaneous estimation of two waveforms is required. Such estimation can only be done jointly with detection of the desired (digital) signal. A useful implementation of such a joint estimator/detector will be shown which employs "Decision-Direction" of the interference-cancelling estimators. In general, only recursive, state-variable filters with resettable states can satisfy such requirements. Thus, a second reason why cancellation techniques have not been applied to the aeronautical problem, other than lack of theory, is lack of hardware and device technology to support such theory. Estimation of wide bandwidth interfering signals requires wide-bandwidth resettable filters. To date, such filters have only been used for relatively narrow band filtering. This is because implementations have used digital logic, with its inherent processing speed limitations. It is only with the advent of new signal processing device technology, such as the Charge-Coupled-Device, and Acoustic-Surface-Wave-Device, that the possibility of wideband interference-cancelling estimation becomes attractive.

The motivation and direction of this investigation, then, is to re-examine the digital detection theory for the aeronautical channel with a

view toward replacing high-cost sub-optimum techniques with more nearly optimum techniques which can take advantage of new low-cost device technology for signal processing.

2.2. THE TECHNICAL APPROACH

The approach for this investigation is, first, to mathematically model the transmitted signal, multipath channel, and additive interfering signals in as general a form as is practically useful. Next, the mathematical models are used to derive detection algorithms for the transmitted signal symbols. These algorithms are to be independent of signal modulation type. The algorithms are to be derived according to particular performance criteria, such as minimum probability of error. Because the algorithms are expected to be algebraically difficult to evaluate in closed form, a Monte Carlo simulation program is to be developed for testing purposes. The simulation is to be capable of evaluating detection algorithm performance over a wide selection of channel conditions, interfering signal types, and transmitted signal types.

The models for the transmitted signal, channel perturbation, and interfering signals are initially models wherein time is a continuous parameter, since this is their physical nature. However, these models are next converted to discrete-time models, for two reasons. First, the simulation must be done in discrete time, since that is the nature of a digital computer. Second, this investigation is initially to obtain optimum interference-cancelling receivers in sampled-data (discrete-time) form. Such algorithms are amenable to implementation using digital logic of C.C.D. device technology. A successful development of interference-cancelling algorithms in discrete-time form would naturally imply an extension of the investigation to continuous-time algorithms.

The continuous-time signal and channel models are developed using complex function notation. There are two reasons for this. First, rough surface scattering theory has been extensively developed using complex notation. Second, the complex signal leads naturally to a two-vector state variable signal formulation where the vector components are the in-phase and quadrature components of the complex signal. The interference-cancelling processor then processes the "I-Q" low-pass components of the received signal.

The mathematical model used for the multipath perturbations is a postulated, or heuristic, one. An exact channel model, derived from first principles of electro-magnetic propagation theory, has never been obtained. Since statistical detection theory is used to derive the optimum detector, an exact channel model is not required. Only a channel model which is statistically equivalent to the multipath perturbations is required. The model used here is of the same type as is used in Sonar work [5], and is a generalization of the model used in previous NASA aeronautical work [6].

The interfering signal model is taken to be a band-pass stochastic process whose carrier frequency is fixed, but arbitrary. Two versions of the signal are available, one in polar form (envelope-phase modulation) and one in quadrature form (quadrature amplitude modulation). Stochastic modulation is employed. By controlling the parameters of the modulation components, any desired type of interfering signal may be realized. For example, using a binary 0-1 sequence in the envelope term and a parabolic sequence in the phase term yields a pulse-modulated, frequency-swept carrier. Using zero-mean Gaussian quadrature component processes yields a noise-modulated carrier.

The desired transmitted signal model is taken first in polar form for generation, and then in quadrature form for reception. Various modulation formats are obtained in the same manner as for the interfering signal.

The technical approach to optimum (interference-cancelling) detection is the following. Based on the mathematical model of the received signal, as perturbed by multipath and interfering signals, the recursive form for the conditional probability density function of received data, given transmitted signal waveform, is computed from the input data. The functions (also loosely called "Likelihood Ratios") are computed separately for each waveform which occurs in the signalling alphabet. The computation begins at the start of a symbol interval and is complete at the end of the same symbol interval. During any symbol period only one of the several possible signal waveforms is actually present in the input data. At the end of the symbol period, the waveform truly present in the input data reveals itself by causing the corresponding density function to be greater than the others which were computed with a false signal waveform assumption. This method of symbol detection is called "Maximum Likelihood."

It is shown in following sections that the detection statistic contains a convex functional of the "Innovations Process" [7]. This process is developed

as the error functions of linear filters which are in feed-back canonical form and are attempting to track the stochastic components of the input data. If it is desired that the operation be optimum from the instant the detection is initiated, or if the stochastic interference is highly non-stationary, then the linear filters must be of the Kalman type. If the interference is nearly stationary and an initial lock-up or learning period can be tolerated, then the linear filters may be stationary, of the Wiener type. Each linear filter is imbedded in the algorithms for computing the Maximum-Likelihood density functions. Each filter is given a different reference waveform, representing the desired signal present. Only the filter having the reference waveform of the signal truly present during the interval "successfully" tracks the stochastic components of the input data. This successful linear filter, produces a tracking error signal (Innovations Process) which is minimum in the mean-squared sense. The filter having minimum error reveals which signal waveform was truly present during the interval.

The stochastic components which the filters attempt to track consist of the additive interfering signal process and any diffuse multiplicative noise due to multipath. The Innovations Process is formed by subtracting the Kalman filter's conditional mean estimates of the stochastic components from the input data stream (predicted estimates in the sampled-data case). Thus, the filter which minimizes the Innovations Process (variance) is the filter which most nearly tracks and, by subtraction, cancels the stochastic interferences. Therefore, the optimum detection strategy (Recursive Maximum Likelihood) is the cancellation strategy.

In the manner of state-variable-type estimators, the optimum detector requires many pieces of subsidiary information. Chief among these are knowledge of the bandwidths and strengths of the interfering signal and of the diffuse multipath perturbations. Also required is knowledge of the strenth of desired signal and additive white noise. Finally, synchronization of the detector with the received symbol interval is required.

It should be noted that, by assuming a general (arbitrary) modulation waveform, spread-sprectrum modulation is not assumed, A Priori. Thus, initially the synchronization problem is only that of synchronizing with a waveform whose time variation is commensurate with that of the information rate itself. Such inherently narrow-band modulation is all that is required,

unless the additive colored interference waveforms are likely to be highly correlated with the desired signal waveforms. In such a case some resort to signal coding of symbols and waveforms may need to be made. This is dealt with below. However, A Priori wide-band spreading of the transmitted signal is not necessarily required.

Because synchronization and the various statistics of desired and interfering signals are not known, A Priori, they must be estimated. These estimations need not be made during a single symbol interval, but may be made over many intervals. In general, the estimates are also made according to the Maximum Likelihood principle. That is, for each statistic, an array of several differing values is tested against the input data, using conditional probability density functions. The array is shifted in a systematic manner to search for the true value of the statistic which is present. Once its true value is bracketed by the array, periodic testing, with shifting of the array keeps the estimate of the statistic up-dated.

The necessity to obtain estimates of the various required statistics before the interference-cancelling detector can be operated in its optimum fashion dictates that the receiver go through a "learning" phase prior to good data detection. Having learned or "acquired" the initial values of the various statistics of the communication environment, the receiver then operates in the interference-cancelling mode, meanwhile "adapting" to changes in the environment statistics. Thus, the interference-cancelling receiver must also employ the learning and adaptive features.

Because of the Integrated Detection Estimation, and Identification (IDEI) employed by the processing algorithms, closed form analytic evaluation of algorithm performance is difficult except in some simplified cases. Thus, Monte Carlo evaluation of the algorithms is a necessity. Properly executed, the simulation is useful for developing and simplifying individual algorithms, as well as for testing overall performance.

The remainder of this report consists of five more major sections. Section III, following, produces the mathematical models for the channel and interferences, culminating in the actual received sampled-data waveform and an equivalent canonical state-variable model.

Section IV presents the derivation of the discrete-time, recursive IDEI algorithms. A simplified example is explained. Then previous related results

from the literature are covered. Next are presented the linear tracking algorithms and the standard detector algorithms for comparison. Finally, some work on signal design and coding is presented.

Section V considers the problem of identifying the statistics of the interfering processes as well as some unknown parameters. This section examines previous work in the area and defines needed work in the follow-on extension to the present contract.

Section VI gives details on the Monte Carlo simulation program. The main routine and subroutines are documented to the minimum extent necessary to understanding and running the simulation. Also, some preliminary simulation results are given.

The final major portion of the report consists of Appendices and References.

SECTION III

SIGNAL AND CHANNEL MODELING

3.1 THE CONTINUOUS-TIME REAL AND COMPLEX MODELS

The transmitted signal is taken in complex function form as

$$\sigma_t(t) = m_t(t) \exp(j\omega_c t) \qquad (3.1.-1)$$

where $m_t(t)$ is a complex, low-pass "modulation" function and $\exp(j\omega_c t)$ is the complex form for the unmodulated "carrier" wave. The constant, ω_c, is the carrier frequency in radians per second. The modulation function is written in terms of a real "envelope" function, $A_t(t)$, and "phase" function, $\phi_t(t)$, as

$$m_t(t) = A_t(t) \exp(j\phi_t(t)); \quad 0 \leq A_t(t) \qquad (3.1.-2)$$

The physical, or realizable, transmitted signal, $s_t(t)$, is taken as the real part of the complex $\sigma_t(t)$.

$$s_t(t) = \text{Re}\ \{\sigma_t(t)\}$$

$$= A_t(t) \cos\ (\omega_c t + \phi_t(t)) \qquad (3.1.-3)$$

Any conceivable modulation format may be represented by a suitable choice of the functions $A_t(t)$ and $\phi_t(t)$. For example, if $A_t(t)$ is unity and $\phi_t(t)$ is non-zero, the signal is phase (or frequency) modulated. If $\phi_t(t)$ is zero and $A_t(t)$ is non-zero, the signal is envelope (or amplitude) modulated. Both the above cases yield signals whose power spectrum is even-symmetric with respect to the carrier frequency. If $A_t(t)\cos\phi_t(t)$ and $A_t(t)\sin\phi_t(t)$ are a Hilbert Transform pair, then the signal, $s_t(t)$, has single-sideband structure and the complex modulation function, $m_t(t)$, is analytic. Intermediate (or residual sideband) cases are also possible. (See Reference [8]).

The complex function which is received at the vehicle is denoted $\zeta(t)$. It is the sum of the transmitted signal, as received over direct line-of-sight path and reflected path, plus an interfering process, $\sigma_j(t)$, plus a white noise

process, $\eta(t)$. The received function is normalized in amplitude with respect to the transmitted signal $\sigma_t(t)$, to write

$$\zeta(t) = \sigma_t(t) + \sigma_r(t-\Delta) + \sigma_j(t) + \eta(t) \qquad (3.1.-4)$$

In (3.1.-4) $\sigma_t(t)$, the transmitted signal, is taken as the component of signal received via the line-of-sight path. $\sigma_r(t-\Delta)$ is the component of desired signal received via a set of paths reflected from the Earth's surface. (See Figure 3.1.-1). $\sigma_r(t-\Delta)$ is normalized in time by the positive constant, Δ, which represents the minimum path differential delay time through the specular reflection point. (See Ref. [6]).

The received function, $\zeta(t)$, is written in polar form as

$$\zeta(t) = [m_t(t) + m_r(t-\Delta) + m_j(t) + \delta(t)] \exp(j\omega_c t) \qquad (3.1.-5)$$

where $m_t(t)$, $m_r(t)$, and $m_j(t)$ are the complex modulation functions (sometimes called complex "envelopes" - Ref. [8]) for the transmitted, reflected, and interfering signals, respectively. Equation (3.1.-5) normalizes frequency with respect to ω_c, which is taken to be the carrier frequency of the transmitted signal, $\sigma_t(t)$, as received via the line-of-sight path. Thus, ω_c includes the direct path Doppler shift between transmitter and vehicle. In general, the reflected component, $m_r(t)$, has slightly different Doppler offset than the direct path. This must be accounted for in formulation of the $m_r(t)$ modulation function, as the function of the transmitted $m_t(t)$. Also, the carrier frequency of the interfering signal, $\sigma_j(t)$, is not generally received equal to ω_c. Thus, the modulation function, $m_j(t)$, is generally one which yields non-symmetric spectra with respect to ω_c. The function, $\delta(t)$, represents complex white Gaussian noise in "baseband" form.

By splitting the modulation functions into their real and imaginery parts, the physical form of the received function may be written as

$$z(t) = \text{Re}\ \{\zeta(t)\}$$

$$= \zeta_i(t) \cos \omega_c t - \zeta_q(t) \sin \omega_c(t) \qquad (3.1.-6)$$

where

$$\zeta_i(t) = y'_{ti}(t) + y'_{ri}(t) + y'_{ji}(t) + n_i(t)$$

$$\zeta_q(t) = y'_{tq}(t) + y'_{rq}(t) + y'_{jq}(t) + n_q(t) \qquad (3.1.-7)$$

and

$$y'_{ti}(t) = \text{Re}\{m_t(t)\} \quad ; \quad y'_{tq}(t) = \text{Im}\{m_t(t)\}$$

$$y'_{ri}(t) = \text{Re}\{m_r(t)\} \quad ; \quad y'_{rq}(t) = \text{Im}\{m_r(t)\}$$

$$y'_{ji}(t) = \text{Re}\{m_j(t)\} \quad ; \quad y'_{jq}(t) = \text{Im}\{m_j(t)\}$$

$$n_i(t) = \text{Re}\{\delta(t)\} \quad ; \quad n_q(t) = \text{Im}\{\delta(t)\} \qquad (3.1.-8)$$

Equations (3.1.-6) - (3.1.-8) define the physically received data process, as it exists in bandpass form at the frequency ω_c in in-phase, quadrature (I-Q) form. This is the form which is convenient for derivation and simulation of the optimum detection algorithms. However, before approaching those tasks, it is necessary to do some detailed modeling of the modulation functions for the transmitted, reflected, and interfering signals.

3.2. THE TRANSMITTED SIGNAL MODELS

The envelope, phase function pair, $A_t(t)$, $\phi_t(t)$, are sufficient to cover any desired modulation format. All symmetric-sideband amplitude modulations are obtained by setting $\phi_t(t) = 0$. All symmetric-sideband angle modulations (FM, PM) are obtained by setting $A_t(t) = 1$. Single-sideband amplitude modulations are obtained by setting

$$A_t(t) \sin \phi_t(t) = \pm H\{A_t(t) \cos \phi_t(t)\} \qquad (3.2.-1)$$

where H{} is the Hilbert transform and the $\pm$ signs give upper and lower sidebands, respectively. Single sideband angle modulations are obtained by setting

$$A_t(t) = \exp\{-H\{\phi_t(t)\}\} \qquad (3.2.-2)$$

Of particular interest to this investigation are the digital modulations. For an M-ary alphabet, a code waveform, c(t), is used, which takes on only M values and switches values at well-defined switching times. Also, an alphabet parameter, m, is used to relate the value of the code to the member of the alphabet. Thus, for the binary, ternary, and quaternary cases, on the time interval, [0, T]

$$c(t,m) = -1 \quad ; \quad m = 0 \quad ; \quad t \in [0,T], \quad m \in \{0,1\}$$

$$c(t,m) = +1 \quad ; \quad m = 1$$

$$c(t,m) = -1 \quad ; \quad m = 0 \quad ; \quad t \in [0,T], \quad m \in \{0,1,2\}$$

$$c(t,m) = \ 0 \quad ; \quad m = 1$$

$$c(t,m) = +1 \quad ; \quad m = 2$$

$$c(t,m) = -2 \quad ; \quad m = 0 \quad ; \quad t \in [0,T], \quad m \in \{0,1,2,3\}$$

$$c(t,m) = -1 \quad ; \quad m = 1$$

$$c(t,m) = +1 \quad ; \quad m = 2$$

$$c(t,m) = +2 \quad ; \quad m = 3 \qquad (3.2.-3)$$

Using the code waveform c(t,m), a binary phase-shift-keyed (PSK) signal is obtained as

$$A_t(t) = 1$$

$$\phi_t(t) = \pi/2 \ c(t,m) \qquad (3.2.-4)$$

Binary FSK is obtained as

$$A_t(t) = 1$$

$$\phi_t(t) = \Delta\omega_t \cdot c(t,m) \cdot t \qquad (3.2.-5)$$

where $\Delta\omega_t$ is the desired frequency shift in radians per second. The quaternary amplitude-minimum-shift-keyed signal used by airlines [34] is obtained as

$$A_t(t) = 1 + a \cdot \sin[c(t,m) \cdot \frac{\pi}{T} t] \qquad (3.2.-6)$$

where a is modulation index and T is digit duration in seconds.

Given the alphabet symbol, $m \in \{0,1,2,\ldots,M-1\}$, the set $c(t,m)$, $A_t(t)$, $\phi_t(t)$ is sufficient to write the complex modulation function, $m(t)$, for any M-ary digital signal.

3.3. THE REFLECTED SIGNAL MODEL

Appendix A gives a qualitative description of multipath propagation due to surface reflection and reviews published research in reflection modeling for the period from 1956 to 1976. Summarizing the results of the past twenty years, there still does not exist an exact reflection model for modulated signals between moving terminals, derived from first principles of electromagnetic (e.m.) propagation theory.

The model which is used in this investigation is heuristic, or postulative, as are all other multipath models to date. It is quite similar to Van Tree's "Doubly-Spread" model which was postulated for the under-water acoustic channel [5]. The e.m. reflection channel and the underwater acoustic channel are closely related since the propagation equations for reflection of horizontally polarized e.m. waves are the same as the equations for reflection of acoustic (compression) waves. Thus, many effects encountered in under-water sonar work have analogs in e-m scattering. The model is a special case of those discussed by Bello in [9].

3.3.1. The Densely Tapped Delay-line Model

Figure 3.3.-1 shows the geometry used for the formal derivation of the complex modulation function of the reflected signal. The transmitted signal, $\sigma_t(t)$, is as defined in (3.1.-1). It is propagated to the receiving point via a direct path of length d meters and via sets of paths of lengths $r + \delta_i$ meters,

for i = 1, 2,..., N. The distance, r, is the minimum path length through the specular reflection point. For each i = 1, ..., N, there results a received signal

$$\sigma_i(t) = \alpha_i(t, \delta_i)m(t - \frac{r+\delta_i}{c})\exp[j\omega_c(t - \frac{r+\delta_i}{c})] \qquad (3.3.-1)$$

c = speed of light in meters/sec.

where $\alpha_i(t, \delta_i)$ is a complex reflection coefficient for the set of paths of length $r + \delta_i$. Note that, in general, $\sigma_i(t)$ is received from a locus of points on the rough reflecting surface which all yield a path length of $r + \delta_i$.

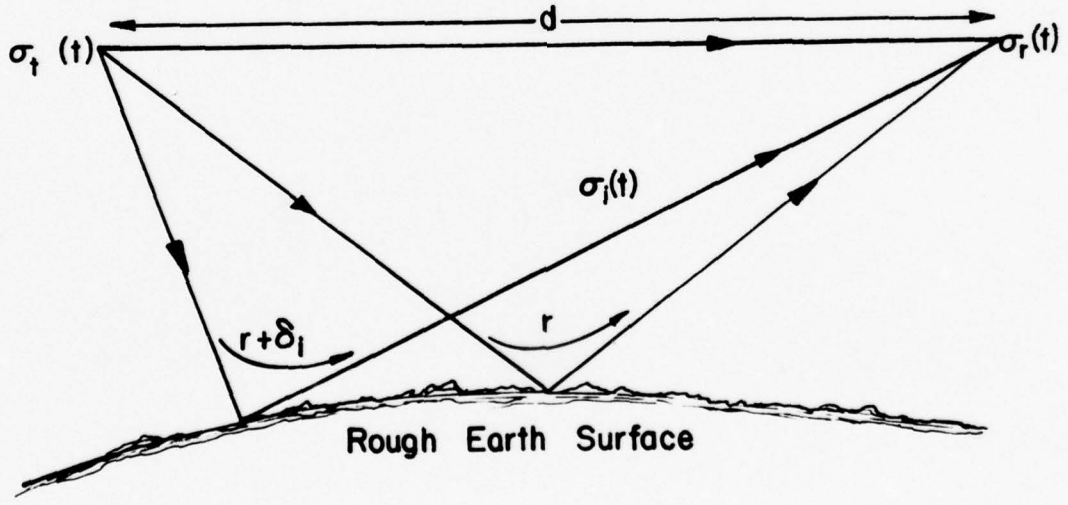


Figure 3.3.-1. Reflection Geometry.

Equation (3.3.-1 is expanded into the form

$$\sigma_i(t) = \rho_i(t, \tau_i)m(t - \tau_r - \tau_i)\exp(-j\omega_c\tau_r)\exp(j\omega_c t) \qquad (3.3.-2)$$

$$\tau_r = \frac{r}{c}, \quad \tau_i = \frac{\delta_i}{c}$$

In (3.3.-2), τ_r is the delay time through the specular point. τ_i is the additional delay time through the ith set of paths. The reflection coefficients, $\rho_i(t, \tau_i)$ for i = 1, 2, ..., are, in general, correlated as τ_j approaches τ_i. This correlation occurs because the locus of points on the reflecting surface yielding τ_j converges to the locus of points yielding τ_i as τ_j approaches τ_i. The variation of $\rho_i(t, \tau_i)$ with t occurs because the ith locus of points moves across the reflecting surface due to aircraft movement, thus changing the net reflection coefficient with time.

The signal received over all paths is

$$\sigma_r(t) = [\sum_{i=1}^{N} \rho_i(t, \tau_i) m_t(t - t_r - \tau_i)] \exp(-j\omega_c \tau_r) \exp(j\omega_c t)$$

$$= [\sum_{i=1}^{N} \rho_i(t, i\Delta T) m_t(t - t_r - i\Delta T) \exp(-j\omega_c \tau_r) \exp(j\omega_c t) \qquad (3.3.-3)$$

where equal-increment delay paths are indicated by setting $\tau_i = i\Delta T$, with ΔT a constant. An equivalent modulation function is defined for the set of reflected paths as

$$m_r(t - t_r) = \sum_{i=1}^{N} \rho_i(t, i\Delta T) m_t(t - t_r - i\Delta T) \qquad (3.3.-4)$$

Note that (3.3.-4) has the interpretation of a tapped delay-line with complex time-varying tap weights, as per Figure 3.3.-2.

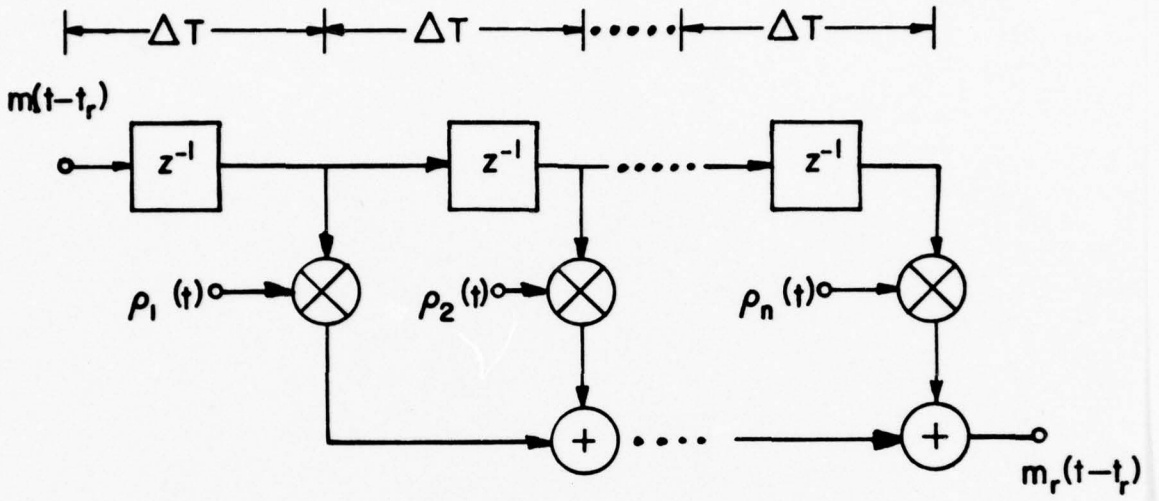


Figure 3.3.-2: Delay-Line Interpretation.

It is assumed that the reflection coefficients are correlated according to

$$E\{\rho_i(t+\tau, \tau_i)\rho_j^*(t, \tau_j)\} = h_i h_j^* R(\tau - (i-j)\Delta T) \qquad (3.3.-5)$$

where $R(0) = 1$ and the h_i are constants. This definition implies that the correlation varies uniformly over the surface. The power associated with a particular set of paths, say the ith, is proportional to $(h_i)^2$. Using (3.3.-5), the autocorrelation function for $m_r(t - t_r)$ is found to be

$$E\{m_r(t+\tau-t_r)m_r^*(t-t_r)\} \triangleq R_{rr}(\tau)$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{N} E\{\rho_i(t+\tau,\tau_i)\rho_j^*(t,\tau_j)\}E\{m_t(t+\tau-t_r-\tau_i)m_t^*(t-t_r-\tau_j)\}$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{N} h_i h_j^* R(\tau-(i-j)\Delta T)R_{tt}(\tau-(i-j)\Delta T) \qquad (3.3.-6)$$

where $R_{tt}(\tau)$ is the autocorrelation function for $m_t(t)$. In (3.3.-6) it is

assumed that the $\rho_i(t, \tau_i)$ and $m_t(t)$ are independent.

Now, (3.3.-6) is just a discrete convolution equation. Thus, as $N \to \infty$ and the partitioning of the sets of reflected paths becomes fine, the autocorrelation function for $m_r(t)$ is formally defined as

$$R_{rr}(\tau) = R(\tau) \cdot R_{tt}(\tau) * h(\tau) * h^*(-\tau) \qquad (3.3.-7)$$

where $h(\tau)$ is an impulse response defined by the h_i. Equation (3.3.-7) implies that for the purpose of deriving the autocorrelation function of $m_r(t)$, an equivalent model may be used

$$m_r(t) = [\rho(t) \cdot m_t(t)] * h(t)] \qquad (3.3.-8)$$

In (3.3.-8), $\rho(t)$ is a complex multiplicative noise process having autocorrelation, $R(\tau)$. The impulse response, $h(t)$, is defined from the density of reflected power versus delay. Note that in the model of (3.3.-4), if the $\rho_i(t, \tau_i)$ are Gaussian then the conditional density of $m_r(t)$, given $m_t(t)$, is also Gaussian, and $m_r(t)$ is characterized by mean and autocorrelation functions only. Thus, (3.3.-8) is an equivalent model, for $\rho(t)$ Gaussian and $h(t)$ deterministic.

Now, the sum of direct-path and reflected-path signals at the receiving point is given by

$$\begin{aligned} \Psi(t) &= \sigma_r(t) + \sigma_t(t - \tau_d) : \tau_d = \frac{d}{c} \\ &= [m_t(t') + m_r(t' - \Delta)\exp(-j\omega_c\Delta)]\exp(j\omega_c t') \\ \Delta &= \tau_r - \tau_d : \text{ Differential Delay Time} \\ t' &= t - \tau_d \end{aligned} \qquad (3.3.-9)$$

If it is assumed that the time (phase) reference is the direct-path signal, then the channel may be modelled as in (3.1.-5) with

$$m_r(t - \Delta) = [\,[\rho(t) \cdot m_t(t - \Delta)] * h(t)]\exp(-j\omega_c\Delta) \qquad (3.3.-10)$$

3.3.2. Special Cases.

i) The Doppler-Spread Case.

When the reciprocal of the delay time across the effective reflection area is large compared to the highest frequency present in the modulation function, the Doppler-Spread case occurs. When the transmitted carrier is unmodulated, there results

$$m_t(t) = 1$$

$$h(t) = \delta(t) \qquad (3.3.-11)$$

and the resulting reflected modulation function is

$$m_r(t) = \rho(t) \qquad (3.3.-12)$$

The complex function, $\rho(t)$, is the Doppler-spreading multiplicative noise function.

ii) The Delay-Spread Case.

When the reflection area moves slowly across the reflecting surface and when the reciprocal of time delay across the area is small compared to the highest frequency present in the modulation function, the Delay-spread case occurs. When the transmitted modulation is a sharp pulse, there results

$$m_t(t) = \delta(t)$$

$$\rho(t) = 1 \qquad (3.3.-13)$$

and the resulting reflected modulation function is

$$m_r(t) = h(t) \qquad (3.3.-14)$$

The real function, $h(t)$, is the Delay-spreading impulse response function.

iii) The Doubly-Spread Case.

In general, the reflected modulation function, $m_r(t)$, is both Doppler-spread and delay-spread. In this case, a time-varying impulse response for the channel is defined by letting the transmitted modulation be a sharp pulse at time $t = t'$,

$$m_t(t) = \delta(t - t') \qquad (3.3\text{-}15)$$

Then, the reflected modulation function is

$$m_r(t) = \rho(t')h(t - t') = h_v(t, t') \qquad (3.3\text{-}16)$$

and $h_v(t, t')$ is the time-varying impulse response.

3.3.3. The Doppler-Spreading and Delay-Spreading Functions.

Because of the manner in which the delay-spreading impulse response, $h(t)$, was defined, it may be determined from the spectrum of power versus delay, when the transmitted sinusoidal signal is unmodulated. That is, if a spectrum $P(\tau)$ is known, where the dimension of P is watts and τ is delay time in seconds with respect to minimum (specular) delay time, then

$$h(\tau) = P^{\frac{1}{2}}(\tau) \qquad (3.3.\text{-}17)$$

Such determinations of $P(\tau)$ are available, either in closed form, as in Bello [9], or as the results of numerical computations, as in Peake [11]. The form $P(\tau)$, as given in Bello [9], is

$$P(\tau) \sim \exp(-a\tau)I_0(b\tau) : 0 \leq \tau$$
$$0 < a, b \qquad (3.3.\text{-}18)$$

where $I_0()$ is the modified Bessel function. The delay power spectra, as computed from Peake's [11] solution appear to decay roughly exponentially. For purposes of designing adaptive receivers, $h(\tau)$ must be modeled using a lumped linear filter having finite numbers of poles and zeroes. Thus, the exponential model seems more desirable.

The complex stochastic multiplicative noise, $\rho(t)$ may be characterized from determinations of the Doppler power spectrum, $S_\rho(\omega)$, when the transmitted sinusoidal signal is unmodulated. Since $\rho(t)$ is modeled as a complex stationary Gaussian process, it is completely characterized by mean and autocovariance function. In general, power spectra $S_{\rho\rho}(\omega)$, may be calculated, as in Peake [11], for the Doppler spectrum as it exists at the radio frequency.

If the multiplicative noise, $\rho(t)$, is defined as

$$\rho(t) = \rho_i(t) + j\rho_q(t) \tag{3.3.-19}$$

and the spectrum $S_\rho(\omega)$ is the spectrum of

$$s(t) = \mathrm{Re}\{\rho(t)\exp[j\omega_c t]\} \tag{3.3.-20}$$

then the equivalent low-pass spectra are

$$S_{\rho_i\rho_i}(\omega) = S_\rho(\omega_c + \omega)U(\omega_c + \omega) + S_\rho(\omega_c - \omega)U(\omega_c - \omega)$$

$$S_{\rho_q\rho_i}(\omega) = j[S_\rho(\omega_c - \omega)U(\omega_c - \omega) - S_\rho(\omega_c + \omega)U(\omega_c + \omega)]$$

$$U(x) = 1 : 0 \leq x$$
$$= 0 : x < 0 \tag{3.3.-21}$$

where the additional properties hold that

$$S_{\rho_i\rho_i}(\omega) = S_{\rho_i\rho_i}(-\omega) = S_{\rho_q\rho_q}(\omega) = S_{\rho_q\rho_q}(-\omega)$$

$$S_{\rho_q\rho_i}(\omega) = S_{\rho_i\rho_q}(-\omega) = -S_{\rho_q\rho_i}(-\omega) \tag{3.3.-22}$$

The covariance functions are then given by the Fourier Inverse Transforms as

$$R_{\rho_i\rho_i}(\tau) = F^{-1}\{S_{\rho_i\rho_i}(\omega)\}$$

$$R_{\rho_q\rho_i}(\omega) = F^{-1}\{S_{\rho_q\rho_i}(\omega)\} \tag{3.3.-23}$$

The multiplicative noise, $\rho(t)$ is zero-mean when the reflection is entirely diffuse. This case obtains at L-band frequencies when the angle between reflecting surface and incident ray is greater than, say, 10° and when the mean surface roughness is large in terms of wavelength. When the reflection has a specular component, such as at VHF, then $\rho(t)$ is non-zero-mean. When the mean is non-zero, it may be taken as real.

3.3.4. The Final Complex Signal Model.

With respect to (3.3.-10), it can be shown that when h(t) represents a filter which has non-zero "d.c. response" ($H(0) \neq 0$), and when the differential delay time is constant or very slowly time-varying, then $m_r(t - \Delta)$ may be represented by

$$m_r(t - \Delta) = [\exp(-j\omega_c\Delta)\rho(t)m_t(t - \Delta)]*h(t) \qquad (3.3.-24)$$

Thus, the final rotation of the complex modulation function given by $\exp(-j\omega_c\Delta)$ in (3.3.-10) is made equivalent to an initial rotation of the complex noise $\rho(t)$. The final complex signal received over the direct and reflected paths is then given by

$$\Psi(t) = [m_t(t) + m_r(t - \Delta)]\exp(j\omega_c t) \qquad (3.3.-25)$$

where $m_r(t - \Delta)$ is given by (3.3.-24).

It should be noted at this point that when the reflection is quasi-specular, the multiplicative noise has a non-zero mean function which is real. Under these conditions the rotation, $\exp(-j\omega_c\Delta)$, acts upon the real mean to produce an equivalent complex mean which is, in general, slowly time-varying. It is this effect which produces the slow envelope fades of the real signal. For completely diffuse reflections, the rotation effects may just be lumped into the stochastic $\rho(t)$, itself.

3.4. THE INTERFERENCE MODELS

The complex formulation of the additive interference signal, $\sigma_j(t)$, of (3.1.-4) is

$$\sigma_j(t) = m_j(t)\exp(j\omega_c t) \qquad (3.4.-1)$$

where $m_j(t)$ is the equivalent complex modulation function for the interference. Note that since ω_c is the radian frequency of the carrier of the desired signal, as received over the direct path, then $\sigma_j(t)$ is explicitly referred to that

frequency. However, in general, the carrier frequency of the interference, ω_j, will not equal ω_c. Thus, the frequency offset of the additive interference must be generated in $m_j(t)$.

In particular, the equivalent modulation function for the interference is written as

$$m_j(t) = A_j(t) \exp[j\ \phi_j(t)] \exp(j\ \Delta\omega_j t)$$

$$\Delta\omega_j = \omega_j - \omega_c \qquad (3.4.-2)$$

where $\Delta\omega_j$ is the radian frequency difference between carrier frequencies of interference and desired signal. $A_j(t)$ and $\phi_j(t)$ are the non-negative envelope function and the phase function, respectively. Then, the in-phase and quadrature, low-pass components of $m_j(t)$ are

$$y'_{ji}(t) = A_j(t) \cos(\Delta\omega_j t + \phi_j(t))$$

$$y'_{jq}(t) = A_j(t) \sin(\Delta\omega_j t + \phi_j(t)) \qquad (3.4.-3)$$

Given the offset frequency, $\Delta\omega_j$, the set $A_j(t)$, $\phi_j(t)$ is sufficient to write the complex modulation function, $m_j(t)$. Deterministic signal structures may be formed, as was done in Section 3.2. for the desired signal. Also, stochastic signal structures may be formed.

It is desirable to model three types of additive interference signals, at this point. The first is a continuous-wave, unmodulated carrier (possibly offset in frequency). The second is a pulse-modulated (on-off) carrier (possibly with swept center frequency). The third is a purely stochastic Gaussian process with power spectrum which is even-symmetric with respect to the (possibly offset) carrier frequency.

The unmodulated carrier is obtained by setting

$$A_j(t) = A_j$$

$$\phi_j(t) = 0 \qquad (3.4.-4)$$

where A_j is a constant which is chosen to set the level of the additive interference, relative to the desired signal.

The pulse-modulated carrier is obtained by setting

$$A_j(t) = A_j \cdot c_j(t, n)$$

$$\phi_j(t) = \tfrac{1}{2} \cdot r_\omega \cdot t^2 \qquad (3.4.-5)$$

where r_ω is the carrier frequency sweep rate in radians per second squared. The function $c_j(t, n)$ is a code waveform, as in Section 3.2., where now

$$\begin{aligned} c_j(t, n) &= 0 \ ; \ n = 0 \\ &= 1 \ ; \ n = 1 \end{aligned} \qquad (3.4.-6)$$

The duration and recurrence rate of the pulses are controlled by a sequence of binary symbols, n, as in (3.4.-6). By a proper choice of the constants, $\Delta\omega_j$ and r_ω, the pulsed carrier may be caused to sweep across the frequency band of the desired signal at any arbitrary sweep rate. Sweeps up in frequency are obtained by choosing the sign of r_ω to be positive or negative, respectively.

The purely stochastic Gaussian process is obtained by generating two independent, zero-mean, low-pass processes, $\rho_{ji}(t)$ and $\rho_{jq}(t)$, having identical auto covariance functions, and setting

$$A_j(t) \cos \phi_j(t) = \rho_{ji}(t)$$

$$A_j(t) \sin \phi_j(t) = \rho_{jq}(t) \qquad (3.4.-7)$$

The exact method for implementing (3.4.-7) and (3.4.-3) is made clear in Section 3.5.

3.5. THE DISCRETE-TIME RECEIVED SIGNAL MODEL.

Equation (3.1.-6) gave the band-pass form of the continuous-time received data process in I-Q form. It is more convenient in notation and in simulation to deal with low-pass, or "baseband," data. So, an I-Q product demodulation is assumed in the form

$$z_i(t) = 2z(t)\cos(\omega_c t + \phi_o(t))$$

$$z_q(t) = -2z(t)\sin(\omega_c t + \phi_o(t)) \qquad (3.5.-1)$$

where it is understood that the resulting terms in frequency $2\omega_c$ are discarded. (3.5.-1) does not necessarily denote a coherent phase demodulation process (coherent amplitude detection), since the arbitrary phase term, $\phi_o(t)$, has been taken in the demodulation reference signals. The I-Q product demodulation translates the I-Q components, $\zeta_i(t)$ and $\zeta_q(t)$, to baseband with a possible phase rotation due to the presence of $\phi_o(t)$.

Thus, define a 2-vector data process, $\underline{z}(t)$, as

$$\underline{z}(t) = \begin{bmatrix} z_i(t) \\ z_q(t) \end{bmatrix} = H_o(t) \cdot \begin{bmatrix} \zeta_i(t) \\ \zeta_q(t) \end{bmatrix}$$

$$= H_o(t)[\underline{y}'_t(t) + \underline{y}'_r(t) + \underline{y}'_j(t) + \underline{n}(t)] \qquad (3.5.-2)$$

where

$$H_o(t) = \begin{bmatrix} \cos\phi_o(t) & \sin\phi_o(t) \\ -\sin\phi_o(t) & \cos\phi_o(t) \end{bmatrix} \quad ; \quad \underline{y}'_t(t) = \begin{bmatrix} y'_{ti}(t) \\ y'_{tq}(t) \end{bmatrix}$$

$$\underline{y}'_r(t) = \begin{bmatrix} y'_{ri}(t) \\ y'_{rq}(t) \end{bmatrix} \quad ; \quad \underline{y}'_j(t) = \begin{bmatrix} y'_{ji}(t) \\ y'_{jq}(t) \end{bmatrix}$$

$$\underline{n}(t) = \begin{bmatrix} n_i(t) \\ n_q(t) \end{bmatrix} \qquad (3.5.-3)$$

Equations (3.5.-2) and (3.5.-3) describe the baseband data vector, provided the receiver did not perform any pre-filtering operation on the signal as it existed at the frequency, ω_c. However, in the event that the optimum

detector operates as a modem, say, at the receiver's intermediate frequency output, such pre-filtering may be present. Thus, equation (3.5.-2) should be modified as

$$\underline{z}(t) = H_o(t)[\underline{h}_e(t) * (\underline{y}'_t(t) + \underline{y}'_r(t) + \underline{y}'_j(t) + \underline{n}(t))] \qquad (3.5.-4)$$

where

$$\underline{h}_e(t) = \begin{bmatrix} h_{ei}(t) \\ h_{eq}(t) \end{bmatrix} \qquad (3.5.-5)$$

is the vector equivalent I-Q impulse response functions, due to pre-filtering. Note that in the general case when the bandpass filter transfer function is not even symmetric with respect to ω_c, then $h_{ei}(t)$ and $h_{eq}(t)$ will be different low pass filter functions. Only in the case where the bandpass filter has even symmetry with respect to ω_c are $h_{ei}(t)$ and $h_{eq}(t)$ identical.

The optimum receiver operates on samples of the received continuous-time waveform. The samples are taken at uniformly spaced time intervals. The basic time reference is taken as the duration, T, of a basic message symbol (bit, in the binary case), as received over the direct path. This implies that the optimum detector has achieved synchronization with the bit timing in the direct-path signal. Ultimately, the detector will employ self-synchronization, using "early-late," Maximum-Likelihood Synchronization.

The sampling rate is taken as a fixed, but arbitrary number, K samples per each signal symbol of length T seconds. Thus, the rate is K/T samples per second. The samples are taken uniformly in time, but symmetrically with respect to the end-points of each symbol interval. This insures that no samples are taken at the end-point of an interval. Thus, the continuous time parameter, t, in the models, is replaced by a discrete time parameter, t_k, where k is sample number and

$$t_k = \left(\frac{k - \frac{1}{2}}{K}\right) \cdot T \quad ; \quad k = 1, 2, \ldots. \qquad (3.5.-6)$$

In derivations and simulation, the actual value of t_k is not important in many of the functions. Thus, wherever possible, t_k, is simply replaced by sample number, k. The actual value of t_k, as given by (3.5.-6), is used, for example, in the amplitude and phase functions, $A_t(t_k)$, $\phi_t(t_k)$.

In order to make the notation more descriptive, the dependence of the transmitted and reflected signal vectors on message symbol, m = 0, 1, 2, ..., M, is specifically noted by letting

$$\underline{y}'_t(k) \longrightarrow \underline{y}'_t(k, m)$$

$$\underline{y}'_r(k) \longrightarrow \underline{y}'_r(k, m) \qquad (3.5.-7)$$

In the event that a non-linear quantizing function, Q{}, is employed, due to an analog to digital convertor, the discrete-time version of equation (3.5.-4) becomes

$$\underline{z}(k, m) = Q\{H_o(k)[\underline{h}_e(k)*[\underline{y}'_t(k, m) + \underline{y}'_r(k, m) + \underline{y}'_j(k) + \underline{n}(k)]]\} \qquad (3.5.-8)$$

where (*) denotes discrete convolutions.

It should be noted from (3.5.-8) that although the discrete-time data stream is quantized upon entering the detector, the signal and noise processes are generated without quantizing in the computer simulation. Thus, $\underline{y}'_t(k, m)$, etc., have the fidelity inherent in the digital word length of the simulation computer. The coarseness of the quantizing function, Q{}, is fixed, but arbitrary, so as to allow determination of the required quantization level for accurate operation of the optimum detection algorithms.

The direct path signal vector is generated from the algorithms

$$\underline{y}'_t(k, m) = H_t(k, m)\begin{bmatrix} 1 \\ \\ 1 \end{bmatrix} \quad ; \quad H_t(k, m) = \begin{bmatrix} f_t(k, m) & 0 \\ \\ 0 & g_t(k, m) \end{bmatrix}$$

$$f_t(k, m) = A_t(k, m)\cos\phi_t(k, m)$$

$$g_t(k, m) = A_t(k, m)\sin\phi_t(k, m) \qquad (3.5.-9)$$

where $A_t()$ and $\phi_t()$ are implicit functions of the code waveform, $c_t(k, m)$.

The reflected signal is generated from

$$\underline{y}'_r(k; m) = [H_t(k; m; \Delta)\underline{\rho}(k)]*h(k)$$

$$\underline{\rho}(k) = H_\rho(k; \Delta)\,[\underline{y}_r(k) + \underline{\mu}_r(k)]$$

$$H_\rho(k; \Delta) = \begin{bmatrix} \cos(\omega_c\Delta) & \sin(\omega_c\Delta) \\ -\sin(\omega_c\Delta) & \cos(\omega_c\Delta) \end{bmatrix}$$

$$H_t(k; m; \Delta) = \begin{bmatrix} f_t(k; m; \Delta) & -g_t(k; m; \Delta) \\ g_t(k; m; \Delta) & f_t(k; m; \Delta) \end{bmatrix}$$

$$f_t(k; m; \Delta) = A_t(t_k - \Delta; m)\cos\phi_t(t_k - \Delta; m)$$

$$g_t(k; m; \Delta) = A_t(t_k - \Delta; m)\sin\phi_t(t_k - \Delta; m)$$

$$\underline{y}_r(k) = \Lambda_r\underline{X}_r(k)$$

$$\underline{X}_r(k + 1) = \Phi_r\,\underline{X}_r(k) + \Gamma_r\,\underline{w}_r(k) \qquad (3.5.-10)$$

In (3.5.-10), $y'_r(k; m)$ is the reflected signal 2-vector which incorporates the transmitted signal 2 X 2 matrix, $H_t(k; m; \Delta)$, as delayed by the differential delay time, Δ, and the multiplicative noise 2-vector, $\underline{\rho}(k)$. The multiplicative noise, $\underline{\rho}(k)$ is formed by the 2 X 2 rotation matrix, $H_\rho(k; \Delta)$, operating on the sum of a zero-mean stochastic 2-vector, $\underline{y}_r(k)$, plus a deterministic mean function 2-vector, $\underline{\mu}_r(k)$. The zero-mean stochastic 2-vector, $\underline{y}_r(k)$, is generated from a zero-mean, white, Gaussian 2-vector, $\underline{w}_r(k)$, using an N^{th}-order filter structure defined by the set $\{\Gamma_r, \Phi_r, \Lambda_r\}$. Thus, $\underline{y}_r(k)$ is assumed to be Markov-N. The desired covariance function for $\underline{\rho}(k)$ (and $\rho(t)$) is obtained through proper choice of the elements of $\{\Gamma_r, \phi_r, \Lambda_r\}$.

The additive noise 2-vector, $\underline{n}(k)$, is modeled as

$$\underline{n}(k) = c_n \underline{w}_n(k) = \begin{bmatrix} n_i(k) \\ \\ n_q(k) \end{bmatrix} \qquad (3.5.-11)$$

where $\underline{w}_n(k)$ is a zero-mean, white, Gaussian 2-vector of unit variance whose two elements are independent. The constant c_n is chosen to realize the desired ratio of direct-path signal power to white noise power at the radio frequency. In terms of the continuous-time RF model, the real signal and noise are

$$s(t) = A_t(t) \cos[\omega_c t + \phi_t(t)]$$

$$n(t) = c_n n_i(t) \cos\omega_c t - c_n n_q(t) \sin\omega_c t \qquad (3.5.-12)$$

where $n_i(t)$ and $n_q(t)$ are zero-mean, unit variance, white and Gaussian. The signal to white noise power ratio for the direct-path signal only is then

$$\frac{S}{N} \triangleq \frac{E\{s^2(t)\}}{E\{n^2(t)\}} = \frac{E\{\tfrac{1}{2}A_t^2(t)\}}{c_n^2} \qquad (3.5.-13)$$

The constant c_n used in (3.5.-11) is then

$$c_n = \sqrt{\frac{\tfrac{1}{2} E\{A_t^2(t)\}}{\frac{S}{N}}} \qquad (3.5.-14)$$

For angle-modulated signals (PSK, FSK)

$$c_n = \frac{1}{\sqrt{2\,\frac{S}{N}}} \quad : \quad \text{PSK, FSK} \qquad (3.5.-15)$$

The variance matrix for $\underline{n}(k)$ is then

$$V_{nn} = E\{\underline{n}(k)\underline{n}^T(k)\} = c_n^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = c_n^2 \, I_2 \qquad (3.5.-16)$$

The additive interfering signal, $\underline{y}_j(k)$, which is a 2-vector, is generated according to

$$\underline{y}'_j(k) = H_j(k)\underline{\rho}_j(k) \qquad (3.5.-17)$$

where $H_j(k)$ is a rotation matrix due to the frequency offset of the interference.

$$H_j(k) = \begin{bmatrix} \cos(\Delta\omega_j \cdot t_k) & -\sin(\Delta\omega_j \cdot t_k) \\ \sin(\Delta\omega_j \cdot t_k) & \cos(\Delta\omega_j \cdot t_k) \end{bmatrix} \qquad (3.5.-18)$$

where t_k is given by (3.5.-6). The vector, $\underline{\rho}_j(k)$ is generated differently, depending on which of the three additive interference signals of Section 3.4. is desired.

For the unmodulated continuous-wave carrier, or for the pulse-modulated swept-frequency carrier, $\underline{\rho}_j(k)$ is generated according to

$$\underline{\rho}_j(k) = \begin{bmatrix} \rho_{ji}(k) \\ \rho_{jq}(k) \end{bmatrix} \quad ; \quad \begin{aligned} \rho_{ji}(k) &= A_j(k)\cos\phi_j(k) \\ \rho_{jq}(k) &= A_j(k)\sin\phi_j(k) \end{aligned} \qquad (3.5.-19)$$

where

$$\begin{aligned} A_j(k) &= A_j \\ &\quad ; \text{ Unmodulated C.W.} \\ \phi_j(k) &= 0 \end{aligned} \qquad (3.5.-20)$$

and

$$\begin{aligned} A_j(k) &= A_j \cdot c_j(k, n) && \text{Pulse-modulated} \\ &\quad ; \\ \phi_j(k) &= \tfrac{1}{2} r_\omega \cdot t_k^2 && \text{Swept-frequency} \end{aligned} \qquad (3.5.-21)$$

For the purely stochastic Gaussian process, $\underline{\rho}_j(k)$ is generated according to

$$\underline{\rho}_j(k) = \underline{y}_j(k) + \underline{\mu}_j(k)$$

$$\underline{y}_j(k) = \Lambda_j \underline{X}_j(k)$$

$$\underline{X}_j(k+1) = \Phi_j \underline{X}_j(k) + \Gamma_j \underline{w}_j(k) \qquad (3.5.22)$$

In (3.5.-22), the 2-vector, $\underline{\rho}_j(k)$, is generated from the sum of a zero-mean stochastic 2-vector, $\underline{y}_j(k)$, plus a deterministic mean function 2-vector, $\underline{\mu}_j(k)$. The stochastic $\underline{y}_j(k)$ is generated from a zero-mean, white Gaussian 2-vector, $\underline{w}_j(k)$, using the M^{th}-order filter structure defined by the set $\{\Gamma_j, \Phi_j, \Lambda_j\}$. Thus, $\underline{y}_j(k)$, is assumed to be Markov-M. The desired covariance function for $\underline{\rho}_j(k)$ and $m_j(t)$ is obtained through proper choice of the elements of $\{\Gamma_r, \Phi_j, \Lambda_j\}$.

The mean vector, $\underline{\mu}_j(k)$, is included in the model of (3.5.-22) for generality. Practically, the occurrence of such a mean in the interference would imply a component analogous to the specular component in the reflected signal. Such a component could occur, for instance, if the interference contained a carrier component, phase-locked to the direct-path signal.

3.6. THE CANONICAL STATE-VARIABLE MODEL.

The adaptive receiver-processor, which is to implement the detection algorithms, utilizes Kalman filters for tracking and canceling of the stochastic elements of the additive and multiplicative interferences. It is well known that Kalman filters require a state-variable model of the generators of the processes which are to be tracked. Until recently it was assumed that knowledge of the model of the true generator was required. Intuitively, this required wealth of detail did not seem reasonable. Since the stationary Wiener filter, derived from the Orthogonal Projection Property of Gaussian Expectation [13], requires only knowledge of the output covariance functions of the processes, it seems reasonable to expect that such knowledge would also suffice for the Kalman filter.

Anderson and Moore [14] and Son and Anderson [15], in 1971 and 1973, respectively, showed that, in fact, only the process correlation functions are required for construction of the Kalman filter, rather than complete knowledge of the structure of the process generator. Athans [16] had shown in 1967 that the Kalman filter structure was not unique to within a Similarity Transformation on the filter (or generator) states. However, the later results [15, 16] were much stronger in showing that assumption of any generator structure capable of producing the given output covariance function for the process yields the same process tracking error variance in the Kalman filter.

The problem of realizing a generator model which, when driven by white noise, produces a process having a given output covariance function is the problem of "System Identification." This problem is examined in Section V below.

The adaptive receiver-processor, which is to implement the detection algorithms, has no knowledge of the detailed received data model, as given in equations (3.5.-8) - (3.5.-22), except for the waveforms of the transmitted signals, $\underline{y}'_t(k, m)$ for $m = 0, 1, \ldots, M-1$. There is freedom to assume a somewhat simpler model, provided it is statistically equivalent to the true model. The assumed model is that which may generate the sum of the three terms, $[\underline{y}'_t(k, m) + \underline{y}'_r(k, m) + \underline{y}'_j(k)]$.

It is instructive to view the actual structure by which the three terms are generated. Dropping arguments for the sake of brevity,

$$y'_t + y'_r + y'_j = (H_{t\Delta}H_\rho y_r)*h + H_j y_j + H_t \mu_t + (H_{t\Delta}H_\rho \mu_r)*h + H_j \mu_j \qquad (3.6.-1)$$

where $H_{t\Delta}$ represents $H_t(k; m; \Delta)$. Now, H_ρ just represents a slow rotation of the vectors $\underline{y}_r$ and $\underline{\mu}_r$. Since $\underline{y}_r$ is stochastic, H_ρ may be absorbed by the time variation of $\underline{y}_r$, to good approximation. $\underline{\mu}_r$ represents the real mean function for the reflection which is non-zero only if the reflection is quasi-specular. H_ρ may be absorbed into $\underline{\mu}_r$ under the assumption that $\underline{\mu}_r$ represents a complex, slowly time-varying mean. With the above simplifications, the canonical state-variable model, for the received data is inferred as in equation (3.6.-2).

$$\underline{z}(k) = H_o(k)\,[H_y(k)\,\underline{Y}(k;\,m) + H_\mu(k;\,m)\,\underline{\mu}(k) + \underline{n}(k)]$$

$$H_o(k) = \begin{bmatrix} \cos\phi_o(k) & \sin\phi_o(k) \\ -\sin\phi_o(k) & \cos\phi_o(k) \end{bmatrix}$$

$$H_y(k) = \left[\begin{array}{cc|c} 1 & 0 & \\ & & H_j(k) \\ 0 & 1 & \end{array}\right]$$

$$\underline{Y}(k;\,m) = \begin{bmatrix} (H_t(k;\,m;\,\Delta)\underline{y}_r(k))*h(k) \\ \hline \underline{y}_j(k) \end{bmatrix}$$

$$H_\mu(k;\,m) = \left[\begin{array}{c|c|c} H_t(k;\,m) & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & H_j(k) \end{array}\right]$$

$$\underline{\mu}(k) = \begin{bmatrix} \underline{\mu}_t(k) \\ \hline (H_t(k;\,m;\,\Delta)\,\underline{\mu}_r(k)*h(k) \\ \hline \underline{\mu}_j(k) \end{bmatrix}$$

$$\underline{\mu}_t(k) = \mu\begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad ; \quad \underline{\mu}_r(k) = \begin{bmatrix} \mu_{ri} \\ \mu_{rq} \end{bmatrix} \quad ; \quad \underline{\mu}_j(k) = \begin{bmatrix} \mu_{ji} \\ \mu_{jq} \end{bmatrix} \qquad (3.6.\text{-}2)$$

In (3.6.-2), $\underline{y}_r(k)$ and $y_j(k)$ are given by (3.5.-10) and (3.5.-22), respectively. The impulse response, h(k), is produced by a filter operating according to

$$v_o(k) = \Lambda_h \underline{x}_h(k)$$

$$\underline{x}_h(k) = \Phi_h \underline{x}_h(k - 1) + \Gamma_h v_i(k) \qquad (3.6.\text{-}3)$$

The filter is diagrammed in Figure 3.6.-1.

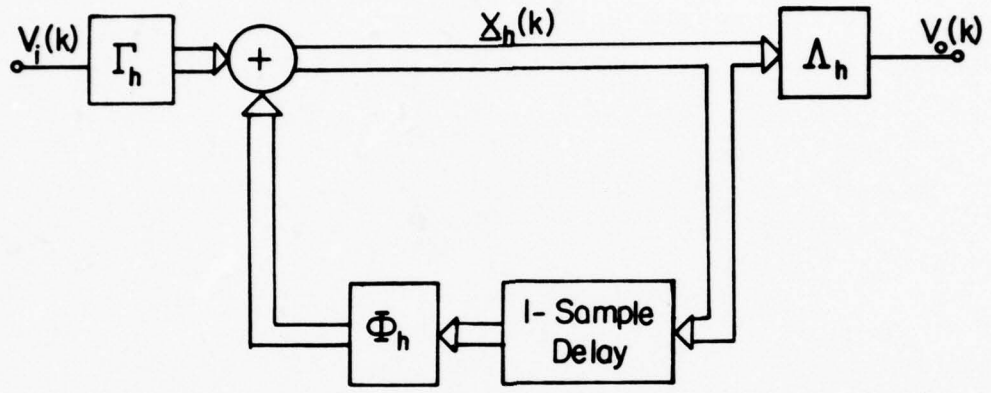


Figure 3.6.-1: Canonical Filter Form.

Note that the filter form of equation (3.6.-3) and Figure 3.6.-1 is not quite the same as that used in the stochastic process generators of equations (3.5.-10) and (3.5.-22). The prior filters produced colored processes from white noise. They inherently produce a one-sample delay from input to output. The filter of equation (3.6.-3) and Figure 3.6.-1 does not produce such a delay between input and output.

The solution to (3.6.-3) is given by

$$v_o(k) = \Lambda_h \Phi_h^k \underline{x}_h(o) + \sum_{n=1}^{k} \Lambda_h \Phi_h^{k-n} v_i(n) \qquad (3.6.-4)$$

The impulse response, h(k), is obtained from (3.6.-4) as follows

$$h(k) = v_o(k); \quad v_i(1) = 1; \quad \underline{x}_h(o) = \underline{0}$$
$$v_i(n) = 0; \quad 1 < n$$

$$= \Lambda_h \Phi_h^{k-1} \Gamma_h; \quad k = 1, 2, \ldots. \qquad (3.6.-5)$$

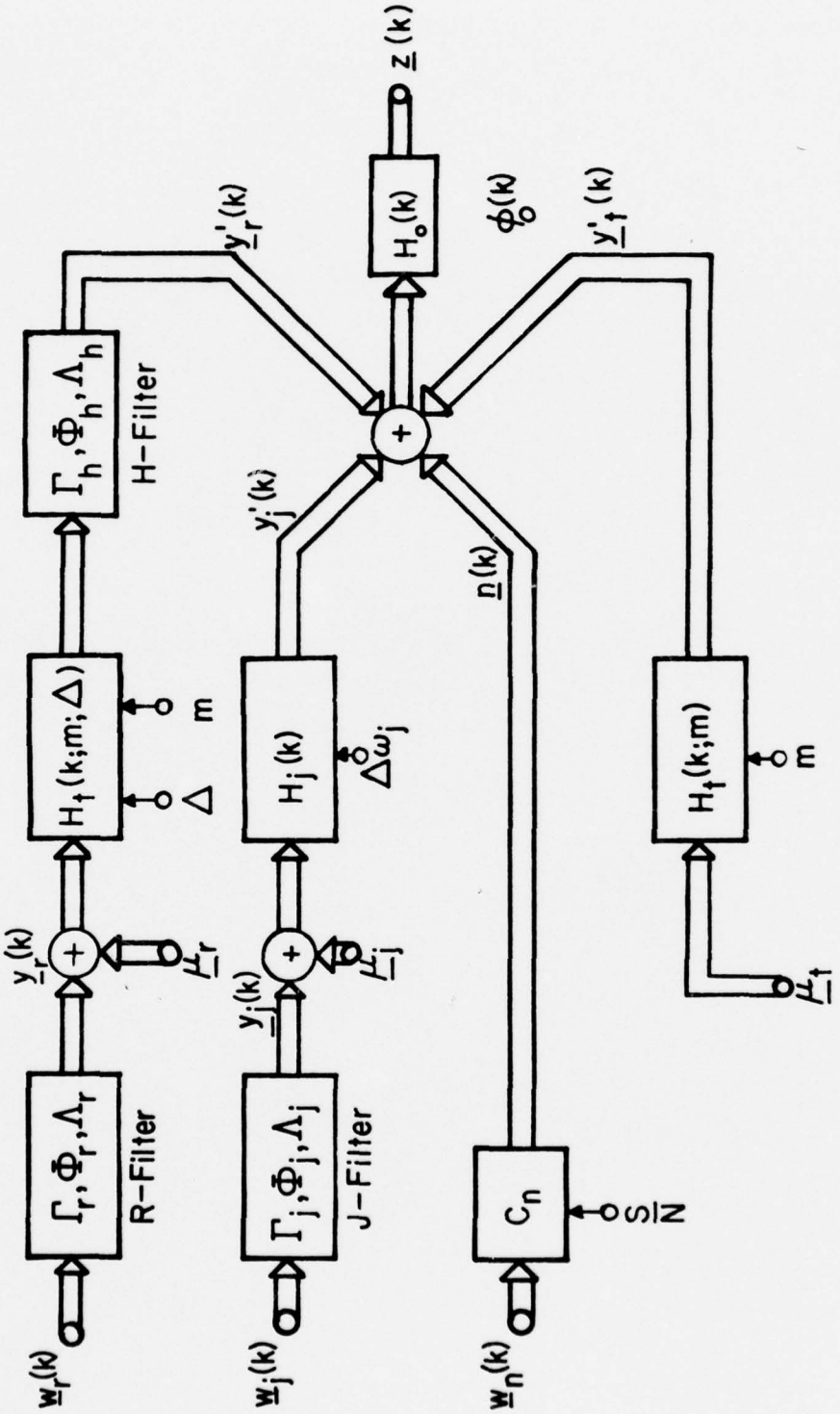


Figure 3.6.-3: Canonical State-Variable Model.

The desired impulse response, h(k) is realized by proper choice of the set $\{\Gamma_h, \Phi_h, \Lambda_h\}$.

The overall configuration of the assumed canonical state variable model is diagrammed in Figure 3.6.-2. The figure combines the elements of equations (3.6.-2), (3.6.-3), (3.5.-22) and (3.5.-10). The upper branch of the model generates the signals and disturbances received via the reflected paths. The Doppler-spreading process, $\underline{y}_r(k)$ is generated from the white noise $\underline{w}_r(k)$, using the R-Filter. The quasi-specular mean, $\underline{\mu}_r(k)$, if present, is added to $\underline{y}_r(k)$ and the sum is multiplied by the modulation matrix, $H_t(k; m; \Delta)$, which is delayed by the process differential delay, Δ. The resulting process is then passed through the Delay-spreading H-Filter.

The second branch of the canonical model generated the colored additive noise process, $\underline{y}_j(k)$, by passing white noise, $\underline{w}_j(k)$, through the J-Filter. The quasi-specular mean, $\underline{\mu}_j(k)$ of this interfering signal, if present, is added to $\underline{y}_j(k)$. The sum is then multiplied by the rotational matrix, $H_j(k)$. Note that for modeling purposes, the additive colored disturbance may be taken as stochastic, even though specific occurrences may be deterministic, as in Section 3.4. This modeling practice is commonly used.

The third branch of the canonical model generates the white additive noise. The level of the white noise, $\underline{w}_n(k)$, is simply adjusted to realize the desired signal to noise ratio, S/N, as in (3.5.-15). The fourth branch of the model generates the direct-path signal. An assumed mean vector, $\underline{\mu}_t(k)$, is multiplied by the modulation matrix, $H_t(k; m)$, to form the signal. The outputs of the four branches are summed and the sum is multiplied by the rotation matrix, $H_o(k)$, to form the received data vector, $\underline{z}(k)$.

The assumed canonical model of Figure 3.6.-1 contains many parameters which are, A Priori, unknown. First among these are the parameters, $\{\Gamma, \Phi, \Lambda\}$, for the R-Filter, H-Filter, and J-Filter. In practice, these parameters for the model must either be set according to prior knowledge of the channel behavior or according to measurements made on the true received data, $\underline{z}(k)$. The problem of determining the elements, $\{\Gamma, \Phi, \Lambda\}$, from measurements on $\underline{z}(k)$ is the Identification problem, which is dealt with below.

Other elements of the canonical model which are A Priori unknown include the mean value functions, $\underline{\mu}_r$, $\underline{\mu}_j$, and $\underline{\mu}_t$. The function, $\underline{\mu}_r$, is essentially the "strength" of any specular reflection which is present. $\underline{\mu}_j$ is the strength of any component of the colored interference which is phase-locked to the

the desired carrier wave. $\underline{\mu}_t$ is the strength of the direct-path signal component. These, possibly slowly time-varying, signal vectors must also be identified during the detection process.

The remaining unknowns in the canonical model are the scalars, Δ, m, $\Delta\omega_j$, S/N, and $\phi_o(k)$. Δ is the differential delay between the direct path and minimum-time reflected path as per Figure 3.3.-1. m is the transmitted symbol (integer), which is to be detected. $\Delta\omega_j$ is the offset frequency between the desired carrier frequency, ω_c, and the interfering carrier frequency, ω_j, as per equation (3.4.-2). S/N is the power ratio of direct-path signal to additive white noise, as per equation (3.5.-13). $\phi_o(k)$ is a possibly slowly time-varying phase perturbation process which is generated in the data receiver in the I-Q product detection of the received radio-frequency waveform. The identification treatment of these scalar processes is detailed below.

There is another A Priori unknown element, which is not explicitly modeled, either in equations (3.5.-10), (3.5.-22), (3.6.-2), (3.6.-3) or in Figure 3.6.-2. This element is denoted τ and represents the timing parameter of the symbols, m, as received over the direct path. The sample timing of equation (3.5.-6) and the following timing of the detector is dependent on bit synchronization with the direct path signal. In the present report, synchronization is assumed. While the study of a self-synchronizing detector is not any more difficult theoretically, the simulation of such is considerably more detailed. Ultimately, the self-synchronization feature should be dealt with.

SECTION IV

RECURSIVE INTEGRATED DETECTION, ESTIMATION, AND IDENTIFICATION

4.1 MAXIMUM A POSTERIORI PROBABILITY DETECTION

4.1.1. The Recursive, Decision-Directed Algorithms

The problem of M-ary detection in channels subject to colored additive and multiplicative noises, as well as additive white noise, is examined below. Detection algorithms are obtained, based on a sub-optimal simplification of the Maximum A Posteriori Probability (MAP) strategy for symbol sequences, using recursive sampled-data processing. The resulting detection algorithms require simultaneous estimation of the colored channel disturbance waveforms, as well as identification of the statistics governing the channel model. The estimation and identification features are, thus, integrated within the detection algorithms.

Early theoretical work on detection in channels perturbed by more than just additive white Gaussian noise [17, 17, 19] led to such developments as the "RAKE" receiver [20] and diversity reception [21]. The idea of adjusting the detector to changing channel conditions led to adaptive detection [22, 23], wherein channel parameters are estimated. An idea dual to adaptive detection, that of estimating the waveform of a signal whose presence is uncertain, was explored in [24, 25]. A related idea, that of differentiating between several possible signals, and simultaneously estimating some signal parameters was explored in [26, 27]. The first recursive sampled-data algorithms for M-ary detection in colored multiplicative noise and white additive noise, using the MAP strategy, were presented in [28]. Simulation results for those algorithms, plus an ad hoc treatment of the required identification problem were given in [29].

The present work extends that of [28] and [29] to include colored additive noise along with colored multiplicative noise, and white additive noise. Also, the identification problem is formally imbedded into the detection/estimation problem by applying the composite detection strategy. The resulting formal solution to the integrated detection/estimation/identification (IDEI) problem

extends the "Marginal Estimation" approach of [30] to the detection problem and employs decision-directed estimation [21] to combat the problem of exponentially growing processor memory. The formal solution presented in this report is a sub-optimal one based on assumed availability of sufficiently good identification estimates. For identification estimates not satisfying the assumption, an extended solution based on the "partitioning" approach of [16] is indicated.

The general M-ary detection problem requires the consideration of detection of a sequence of symbols (or data word) of arbitrary length. Thus, the symbol notation, m, introduced in equation (3.2.-3), is subscripted to indicate the position of each symbol in a sequence. In particular, consider the detection of a J-sequence of message symbols, $m_1, m_2, \ldots, m_J$, where the m_j are elements of the M-ary alphabet,

$$m_j \in \{0, 1, 2, \ldots, M\text{-}1\} \quad \forall_j = 1, 2, \ldots, J \qquad (4.1.\text{-}1)$$

The m_j are imbedded in a sequence of received vector-data samples, $\underline{z}(k)$, for sample times, k = 1, 2, 3.... The number of components of the $\underline{z}()$ vector is determined according to the assumed data generating model, discussed above. A fixed number, K, of data vectors, $\underline{z}()$, are measured for each transmitted message symbol, m_j. Thus, the data sample number, k, and message symbol number, j, are related by

$$j = 1 + \text{Int}\left(\frac{k - \frac{1}{2}}{K}\right) \quad : \quad \text{Int}\,() \text{ is "integer part"} \qquad (4.1.\text{-}2)$$

Under the Maximum A Posteriori Probability (MAP) decision strategy, decision statistics, S', are computed for each of the M^J possible message sequences. The statistics, S', are proportional to the A Posteriori Probability of message sequence, given the received data sequence.

For the purpose of forming the decision statistics, two vectors are defined.

$$\underline{M}_J = [m_J, m_{J-1}, \ldots, m_2, m_1]^T = \begin{bmatrix} m_J \\ \hline \underline{M}_{J-1} \end{bmatrix}$$

$$\underline{Z}_J = [\underline{Z}_J, \underline{Z}_{J-1}, \ldots, \underline{Z}_2, \underline{Z}_1]^T = \begin{bmatrix} \underline{Z}_J \\ \hline \underline{Z}_{J-1} \end{bmatrix} \qquad (4.1.\text{-}3)$$

In (4.1.-3), the m_j are the individual symbols in the message J-sequence. The $\underline{Z}_j$ are vectors of received vector-data samples corresponding to each message symbol, m_j. For K samples per symbol, each $\underline{Z}_j$ vector contains K of the $\underline{z}()$ data vectors. Figure 4.1.-1 illustrates the partitioning for scalar data $\underline{z}()$, for a binary symbol sequence.

The individual decision statistics are all of the form

$$S' \sim p(\underline{M}_J | \underline{Z}_J) \qquad (4.1.-4)$$

where p() denotes the conditional probability density. The MAP decision rule is to choose the particular symbol sequence $\hat{\underline{M}}_J$ which maximizes the density p().

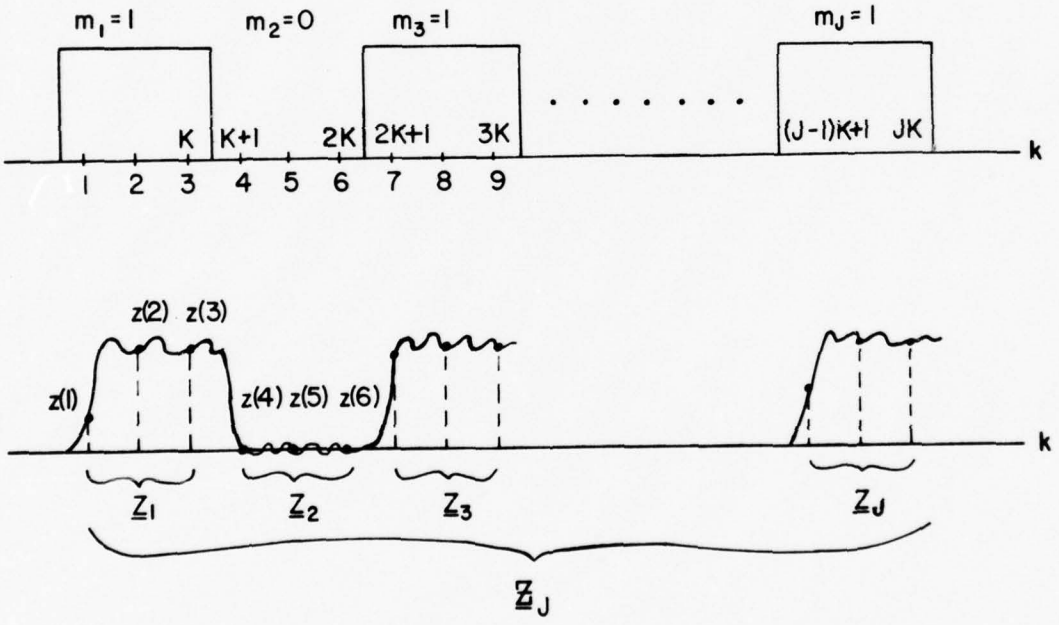


Figure 4.1.1: Partitioning of Sampled-Data Vector

It is assumed that there is associated with the data process, $\underline{z}(k)$, another process $\underline{\beta}(k)$, which in a manner to be specified below, represents some unknown elements of the generator producing $\underline{z}(k)$. The vector process, $\underline{\beta}(k)$, is assumed to be of finite dimension. In the same manner as in (3) above, the $\underline{\beta}()$ samples are grouped to form a partitioned vector, $\underline{\beta}_J$, where

$$\underline{\mathcal{B}}_J = [\underline{B}_J, \underline{B}_{J-1}, \ldots, \underline{B}_2, \underline{B}_1]^T = \begin{bmatrix} \underline{B}_J \\ \hline \underline{\mathcal{B}}_{J-1} \end{bmatrix} \qquad (4.1.-5)$$

In (4.1.-5), $\underline{B}_j$ contains K of the $\underline{\beta}()$ sample vectors which occur during the time of the jth message symbol.

A joint density of $\underline{M}_J$, $\underline{\mathcal{B}}_J$, and $\underline{\mathcal{Z}}_J$ is now postulated. Then, the decision statistic is written formally as

$$S' \sim \frac{p(\underline{M}_j, \underline{\mathcal{Z}}_J)}{p(\underline{\mathcal{Z}}_J)} = \frac{\int \ldots \int p(\underline{M}_J, \underline{\mathcal{B}}_J, \underline{\mathcal{Z}}_J) d\,\underline{\mathcal{B}}_J}{p(\underline{\mathcal{Z}}_J)} \qquad (4.1.-6)$$

$$S' = \int \ldots \int p(m_J, \underline{B}_J, \underline{Z}_J | \underline{M}_{J-1}, \underline{\mathcal{B}}_{J-1}, \underline{\mathcal{Z}}_{J-1}) d\underline{B}_J \; p(\underline{M}_{J-1} | \underline{\mathcal{B}}_{J-1}, \underline{\mathcal{Z}}_{J-1}) \cdot$$

$$\cdot \; p(\underline{\mathcal{B}}_{J-1} | \underline{\mathcal{Z}}_{J-1}) \; d\underline{\mathcal{B}}_{J-1} \qquad (4.1.-7)$$

where the multiple integration is over all the unknown elements of process $\underline{\beta}()$, at every sample time, k. The averaging implicit in (4.1.-7) is essentially the application of the composite detection strategy. The practical implementation of (4.1.-7) would require increasing the order of the multiple integral with increasing numbers of data samples and/or increasing length of the message sequence. Because this implies a signal processor of expanding size, such a result is not practically useful. An alternative is to employ decision-directed processing.

Now, it is assumed that at the end of the (J-1)st symbol period, a conditional-mean estimate of $\underline{\mathcal{B}}_{J-1}$ is available as $\hat{\underline{\mathcal{B}}}_{J-1}$. Furthermore it is assumed that both $p(m_J, \underline{B}_J, \underline{Z}_J | \underline{M}_{J-1}, \underline{\mathcal{B}}_{J-1}, \underline{\mathcal{Z}}_{J-1})$ and $p(\underline{M}_{J-1} | \underline{\mathcal{B}}_{J-1}, \underline{\mathcal{Z}}_{J-1})$ do not vary appreciably in the neighborhood of $\underline{\mathcal{B}}_{J-1}$ for which $p(\underline{\mathcal{B}}_{J-1} | \underline{\mathcal{Z}}_{J-1})$ is significant. Then the integral of the product of the three probability functions in (4.1.-7) simply evaluates the product of the first two at $\underline{\mathcal{B}}_{J-1} = \hat{\underline{\mathcal{B}}}_{J-1}$, analogous to sifting with a Dirac delta function. Next, $p(\underline{M}_{J-1} | \underline{\mathcal{B}}_{J-1}, \underline{\mathcal{Z}}_{J-1})$ reduces to $p(\underline{M}_{J-1} | \underline{\mathcal{Z}}_{J-1})$ since $\hat{\underline{\mathcal{B}}}_{J-1}$ is a function of the $\underline{\mathcal{Z}}_{J-1}$. Then, maximization of S' with respect to $\underline{M}_{J-1}$ is accomplished by choosing the $\hat{\underline{M}}_{J-1}$ which maximizes $p(\underline{M}_{J-1} | \underline{\mathcal{Z}}_{J-1})$. It follows that S' may be maximized sequentially, over each succeeding message symbol, $m_1, m_2, \ldots, m_J$. Thus to reduce the order of the integral in (4.1.-7), decision on the sequence as a whole is replaced with symbol by symbol decision.

At the end of an arbitrary symbol period, say the jth, the cumulative statistic, S', is maximized by maximizing the jth period statistic, S_i, over i = 0, 1, 2, ..., M-1, where

$$S_i = \int \cdots \int p(\underline{B}_j, \underline{Z}_j | m_j = i, \hat{\underline{M}}_{j-1}, \hat{\underline{B}}_{j-1}, \underline{Z}_{j-1}) d\underline{B}_j \; p(m_j = i | \hat{\underline{M}}_{j-1}) \tag{4.1.-8}$$

The statistic S_i is the statistic used for making the decision on the jth symbol. Note that this decision is dependent on previous decisions through $p(m_j|\hat{M}_{j-1})$, in case the transmitted symbols are not independent. The jth symbol decision is also dependent on previous data through the use of $\hat{\underline{B}}_{j-1}$, the sequence of previous identifications of the $\underline{\beta}(k)$ from the sequence of previous data $\underline{z}(k)$.

To pursue the formation of the decision statistic, S_i, during the jth symbol period, it will be helpful to write $\underline{B}_j$ and $\underline{Z}_j$ as explicit functions of the sample number, k. Thus, define

$$\underline{Z}_j = \underline{Z}_j(k) = \begin{bmatrix} \underline{z}_j(k) \\ \hline \underline{Z}_j(k-1) \end{bmatrix}; \quad \underline{z}_j(k) = \underline{z}(jk)$$

$$\underline{B}_j = \underline{B}_j(k) = \begin{bmatrix} \underline{\beta}_j(k) \\ \hline \underline{B}_j(k-1) \end{bmatrix}; \quad \underline{\beta}_j(k) = \underline{\beta}(jk) \tag{4.1.-9}$$

Then, by straight-forward manipulation, (4.1.-8) is placed in the form

$$S_i = \int \cdots \int \prod_{k=(j-1)K+1}^{jK} p(\underline{z}(k), \underline{\beta}(k) | m_j = i, \hat{\underline{M}}_{j-1}, \underline{B}_j(k-1), \hat{\underline{B}}_{j-1}, \underline{Z}_j(k-1, \underline{Z}_{j-1}) \; d\underline{b}_j(k) \cdot p(m_j = i | \hat{\underline{M}}_{j-1}) \tag{4.1.-10}$$

The signal processor implied by (4.1.-10) is still not practical because of the increasing number of integrations as the sampling rate per symbol, K, increases. It is possible, by extending the previous assumptions to do away with the integrals entirely. First, however (4.1.-10) is taken into the following form

$$S_i = \int \cdots \int \prod_{k=(j-1)K+1}^{jK} p(z(k)|m_j=i, \hat{\underline{M}}_{j-1}, \underline{B}_j(k), \hat{\underline{B}}_{j-1}, \underline{Z}_j(k-1), \underline{\mathcal{Z}}_{j-1}) \cdot$$

$$\cdot p(\underline{\beta}(k)|m_j=i, \hat{\underline{M}}_{j-1}, B_j(k-1), \hat{\underline{B}}_{j-1}, \underline{Z}_j(k-1), \underline{\mathcal{Z}}_{j-1}) d\underline{B}_j(k) \cdot$$

$$\cdot p(m_j=i|\hat{\underline{M}}_{j-1}) \qquad (4.1.-11)$$

Now, two more key assumptions are made. Suppose that at sample time k-1 (within the product) and for each i, a conditional-mean estimate for $\underline{B}_j(k-1)$ is available as $\hat{\underline{B}}_j(k-1, i)$. Also suppose that at that time and for each i a conditional-mean prediction is available for $\underline{\beta}(k)$ as $\hat{\beta}(k|k-1, i)$. Provided that both of these estimates have sufficiently small variances, the jth symbol statistic of (4.1.-11) reduces to

$$S_i = \prod_{k=(j-1)K+1}^{jK} p(z(k)|m_j=i, \hat{\underline{M}}_{j-1}, \hat{\underline{\beta}}(k|k-1, i), \hat{\underline{B}}_j(k-1, i), \hat{\underline{B}}_{j-1},$$

$$\underline{Z}_j(k-1), \underline{\mathcal{Z}}_{j-1}) \cdot p(m_j=i|\hat{\underline{M}}_{j-1}) \qquad (4.1.-12)$$

Equation (4.1.-12) represents the practical IDEI detection algorithms under the assumption that suitable conditional-mean identification estimates are available . As the variances of the identification estimates increase it can be expected that the performance of the detection algorithms will degrade. If sufficiently good identification estimates cannot be obtained, then the algorithm of (4.1.-11) should be used, with the integrations performed using the Partitioning Theorem of [32].

When the data, $\underline{z}(k)$, is generated as in equations (3.5.-10), (3.5.-22), and (3.6.-2), then the vector, $\underline{\beta}(k)$ represents all the included parameters which may be A Priori unknown. Thus, $\underline{\beta}(k)$ includes all the structural elements in the sets $\{\Gamma, \Phi, \Lambda\}$ for the R-Filter, H-Filter, and J-Filter. Also included in $\underline{\beta}(k)$ are $\underline{\mu}_r$, $\underline{\mu}_j$, $\underline{\mu}_t$, Δ, $\Delta\omega_j$, and S/N.

Since many of the parameters may be time varying, $\underline{\beta}(k)$ is taken as a stochastic process, rather than just a random vector. In the present case, it is assumed that $\phi_o(k)$ is created in the data receiver. Thus, $H_o(k)$ is measureable and need not be included in $\underline{\beta}(k)$

With the data generated as assumed above and under the above assumptions on the composition of $\underline{\beta}()$, the density, $p(\underline{z}(k)|(\,)\,)$, required in the detection algorithm of (4.1.-12) is conditionally Gaussian, of the form

$$p(\underline{z}(k)|(\,)\,) =$$

$$= \frac{1}{2\pi}[\det V_{\nu\nu}(k|k-1)]^{-\frac{1}{2}}\exp[-\tfrac{1}{2}\underline{\nu}^T(k)V_{\nu\nu}^{-1}(k|k-1)\underline{\nu}(k)]$$

$$\underline{\nu}(k) = \underline{z}(k) - \hat{\underline{z}}(k|k-1)$$

$$\hat{\underline{z}}(k|k-1) = E\{\underline{z}(k)|()\}$$

$$V_{\nu\nu}(k|k-1) = E\{\underline{\nu}(k)\underline{\nu}^T(k)|()\} \qquad (4.1.-13)$$

When $\underline{z}(k)$ is conditioned on the m_j truly present, $\underline{\nu}(k)$ is the Innovations Process [7]. In computing the M statistics S_i, as in (4.1.-12), the true m_j is used in only one of the $\underline{\nu}(k)$. Thus, $\underline{\nu}(k)$ is called here the Pseudo-Innovations. Since $\underline{z}(k)$ is conditionally Gaussian, both $\underline{\nu}(k)$ and $V_{\nu\nu}(k|k-1)$ may be obtained from Kalman filters. In the Kalman filter, $\underline{\nu}(k)$ is the dynamic feedback tracking error, formed in the filter. Each of the M different filters actually attempts to track the colored interference processes, $\underline{y}_r(k)$ and $\underline{y}_j(k)$. Note that in a practical implementation of the receiver, Kalman filters may not be used, due to the computational complexity involved in on-line solution for the Kalman gain function. If it is assumed that the statistics of the interfering processes are, at most, slowly time-varying with respect to symbol time, T, then the processes, $\underline{y}_r(k)$ and $\underline{y}_j(k)$, are almost stationary. Thus, the elements of the various filters in the sets $\{\Gamma, \Phi, \Lambda\}$ are constant over periods of time, long with respect to symbol time, T. Thus, in practical implementations, the Wiener steady-state version of the Kalman filter may be used. To use the Wiener filter in the practical case will require the use of an "acquisition" or "lock-up" mechanization to initialize the filter. This is discussed below, with respect to the Identification problem.

The theoretically optimum receiver is based on the Kalman filter. For purposes of bounding the best possible performance of sub-optimum (Wiener) implementations, the Kalman mechanization is assumed. The physical operation of the optimum detection algorithms is now explained with reference to Figures 4.1.-2 and 4.1.-3.

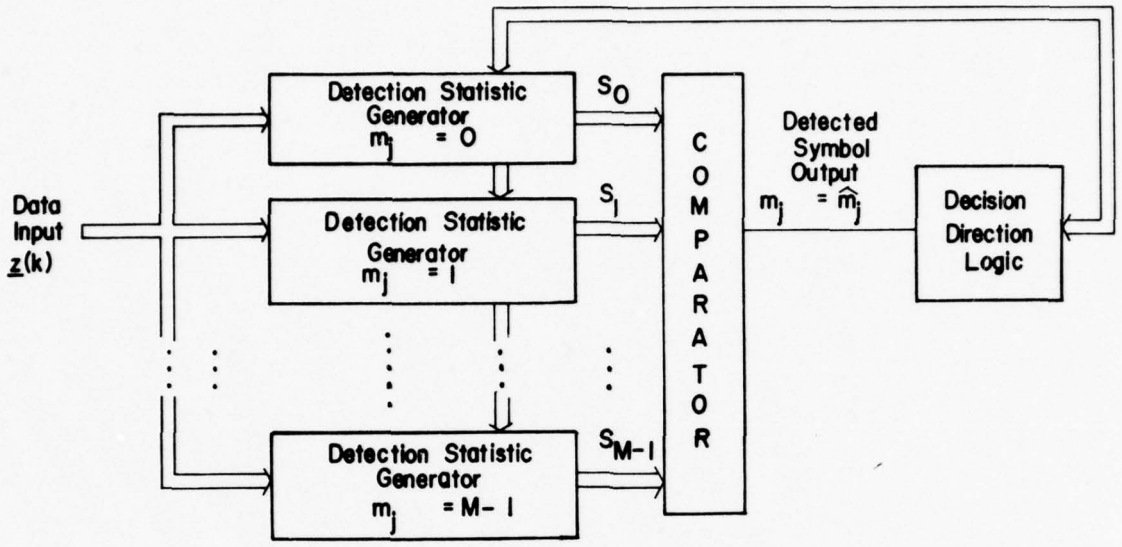


Figure 4.1.-2: Decision-Directed MAP Detector (IDEI)

During the jth symbol period, M unique detection statistics S_i are computed in parallel, one for each of the possibly present m_j. Each separate statistic generator contains its own Linear (Kalman) Filter, Gaussian Function Generator, Conditional-Mean Predictor/Filter, and Product Accumulator. At the end of the jth symbol period, the non-negative statistics, S_i, are compared in magnitude. If S_n is largest, then the decision is made $\hat{m}_j = n$.

At the end of the jth symbol period, the decision direction feature is employed as follows. When the decision is made, $\hat{m}_j = n$, it is inferred that the nth detection statistic generator has been processing the data using the true value of m_j. Thus, it is inferred that the Kalman filter and Conditional-Mean Identifying filter in the nth statistic generator contain good final filtered estimates, $\hat{\underline{y}}_r(jK)$, $\hat{\underline{y}}_j(jK)$, and $\hat{\underline{\beta}}(jK)$, respectively. These final filtered estimates are then routed to the other M-1 statistic generators to reset their initial predicted estimates for the (j+1)st symbol period.

Some observations about the physical operation of the data processor are now in order. First, note that the data model of (3.6.-2) is stochastic at

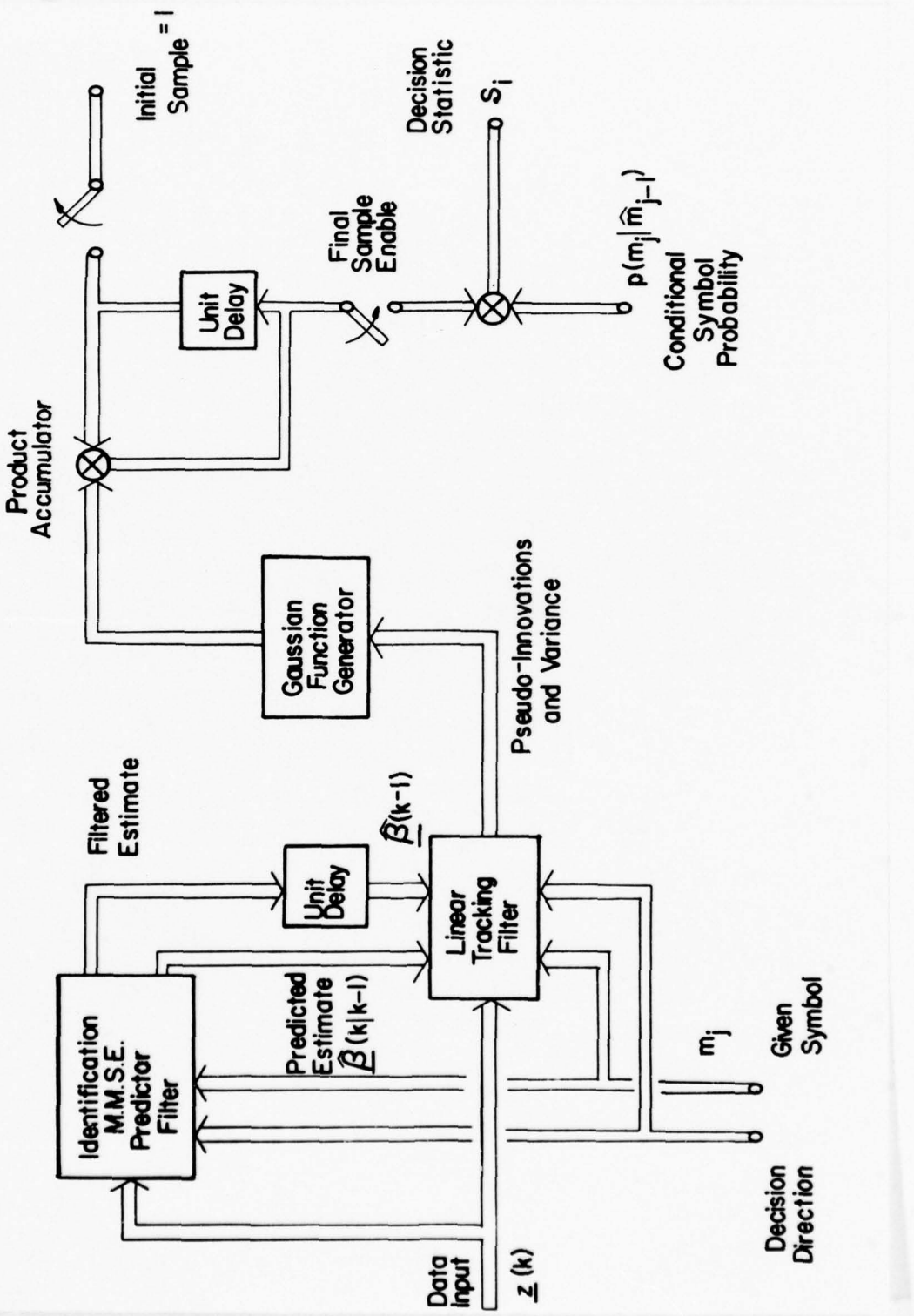


Figure 4.1.-3: Detection Statistic Generator (Fully Adaptive)

two levels. At the first level, stochastic interferences, $\underline{y}_r(k)$ and $y_j(k)$, have been postulated. At the second level, the usually fixed elements of the data generating model, such as mean and Markovian filter structure, have themselves been modeled as stochastic. In the optimum processor, the first-level interference processes are estimated in the linear filter. These processes are actually tracked by the filter. In the formation of the Innovations Process, the interference estimates are subtracted from the incoming data in an attempt to cancel the interferences. It is the conditional-mean predictor/filter which attempts to track the second-level stochastic processes representing the various identification elements of the data model. Note that for those identification elements which admit to Gaussian models, the conditional-mean estimators are also linear. However, for those elements which cannot be modeled as Gaussian, the conditional-mean estimator will be non-linear.

4.1.2. Example

Figures 4.1.-4 and 4.1.-5 relate to a highly idealized example, presented here to clarify some of the preceding ideas. Assume binary phase-shift-keying with $\pm 90°$ phase shift in the presence of an additive colored interference process and white noise. No multiplicative channel disturbance is assumed. Also, postulate phase coherent translation of the band-pass data to baseband, using an unperturbed phase reference (a highly idealized case). Under the assumptions, the desired signal is resident in the quadrature channel only, and the data is scalar, continuous time, taken here as $z(t)$. Instead of Kalman filters, suboptimum stationary Wiener filters may be postulated in the feed-back canonical form of Figure 4.1.-5. If Charge Coupled Device[1] implementation is assumed for these filters, then the conversion from continuous time to discrete time is inherent in the filter structure.

[1]The recursive structure indicated in Figure 4.1.-5 is not the usual CCD transversal filter structure common to the CCD art. The state-variable feedback structure is required so that the states may be reset at the end of each symbol period. The design of such a CCD device is being investigated presently at Texas A&M University.

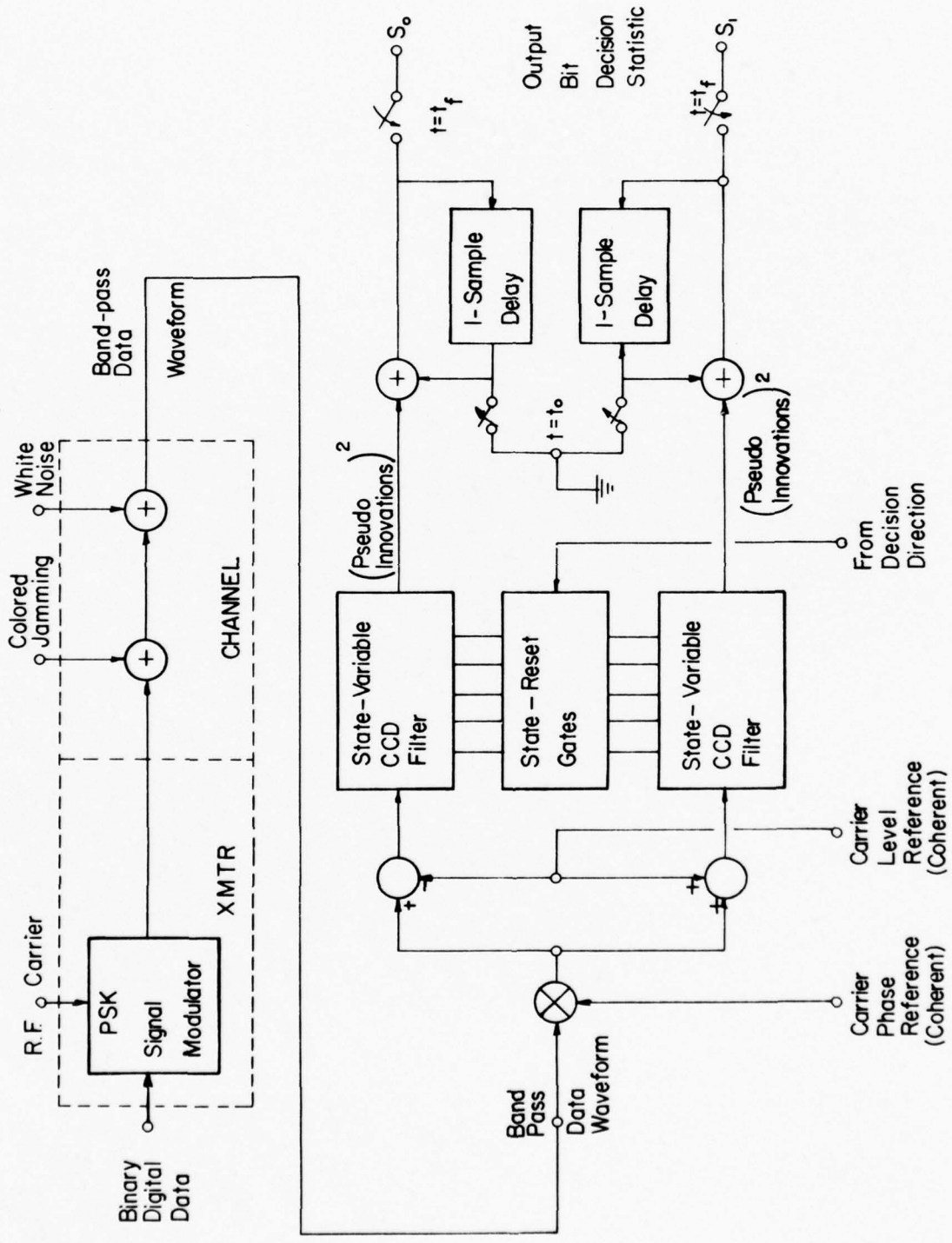


Figure 4.1.-4: Example: Jammer-Tracking PSK Detector (Simplified)

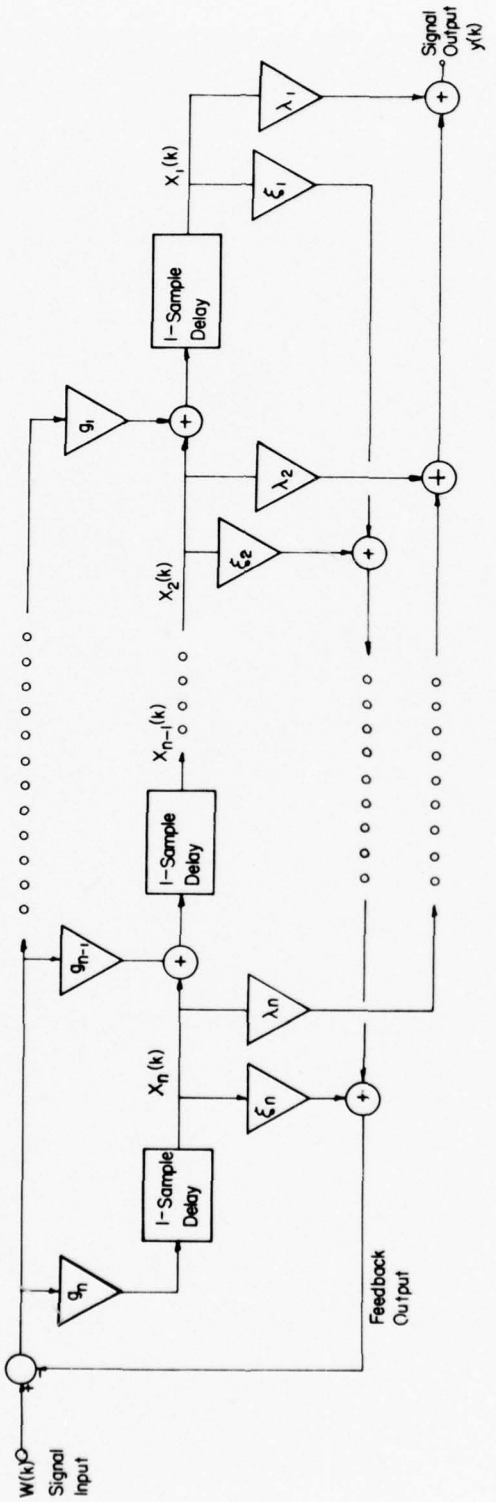


Figure 4.1.-5: Recursive State-Variable CCD Filter

For this example, the components of (3.6.-2) are

$$H_o(k) = 1$$

$$H_Y(k; m_j) = 1$$

$$\underline{Y}(k) = y_j(k) : \text{ a scalar function}$$

$$H_\mu(k; m_j = 0) = +A : 0 < A$$

$$H_\mu(k; m_j = 1) = -A$$

$$\underline{\mu}(k) = 1$$

$$\underline{n}(k) = n(k) : \text{ a scalar function} \qquad (4.1.-14)$$

The required identification for this example includes the carrier reference level, A, the symbol timing (synchronization), and the set of constants in $\{\Gamma(), \underline{\Phi}(), \Lambda()\}$. Identifying $\{\Gamma(), \underline{\Phi}(), \Lambda()\}$ is essentially identifying the power spectrum of the colored additive interference and then synthesizing a suitable minimum-phase recursive filter for tracking $y_j(k)$.

In Figure 4.1.-4, the upper filter is for $m_j = 0$. The lower filter is $m_j = 1$. In the upper channel, $\hat{A}$ is subtracted from the scalar data to produce $y_j(k) + n(k)$ when $m_j = 0$. In the lower channel $(-\hat{A})$ is subtracted from the scalar data to produce $y_j(k) + n(k)$ when $m_j = 1$. Each filter then attempts to track $y_j(k)$ under the differing assumptions on m_j. For this case the sum of the squares of the pseudo-innovations forms a sufficient statistic for detection. Thus, the sum of squares is accumulated recursively using the scheme shown in Figure 4.1.-4. After each symbol decision the final states in the incorrect filter are reset using the final states in the correct filter.

It can be seen from this example that the error properties of these detection algorithms are affected by the transient responses of the linear filters. Thus, closed form expressions for the probability of detection error must be evaluated numerically for particular interference cases. For this reason, the initial approach to algorithm evaluation is through computer simulation as described below. Some closed form results will be available in a subsequent report on Phase II of the present contract.

4.1.3. Previous Related Results.

(i) Lainiotis' Results.

Lainiotis solved a simpler version of the problem in 1969-71 [25]. The problem was detection of a binary "on-off-keyed" (OOK) colored Gaussian process, generated by a given state-variable model. The generating model had unknown (random)imbedded parameters. The detection strategy was "one-shot." That is, only one symbol was to be detected, with no consideration of a sequence of symbols. His work was an extension of previous work by Middleton and Esposito [26], on simultaneous detection and estimation.

Lainiotis' solution was a continuum (over the unknown structural parameters) of likelihood ratios in which were imbedded estimators for the colored signal process. The result was similar to that obtained by Fredriksen, Middleton, and Vandelinde [27] in 1972. The present work of the author extends Lainiotis' work to M-ary alphabets, sequence of symbols, colored interference, and stochastic generating structure. The resulting algorithm is also in a natural form for adaptive implementation. Lainiotis' work culminated in publication of his Partitioning Theorem in 1976 [32]. The partitioning approach to the problem addressed in this report has been mentioned in 4.1.1., above.

(ii) Kailath's Results.

The present results on discrete time detection theory are also quite analogous to some previous continuous-time work of Kailath [33], concerning the Likelihood Ratios (LR) for detection of binary random signals in Gaussian noise. The probelm analyzed by Kailath was initially that of detection of an "on-off-keyed" (OOK) colored stochastic signal in white Gaussian noise. The continuous-time formulation for the LR was

$$LR = \exp\left[\int\!\!\!\!\!-_0^T \hat{z}_1(t)\, \dot{x}(t)dt - \tfrac{1}{2} \int_0^T \hat{z}_1^{\,2}(t)dt \right] \qquad (4.1.-15)$$

where $\dot{x}(t)$ was the observed data process on the time interval $[0,T]$ and $\hat{z}_1(t)$ was the causal conditional-mean estimate of the colored signal, under the additional conditioning that the signal was present. The barred integral denoted the stochastic Ito integral.

There is a close correspondence between the discrete-time detection algorithms of the present paper and those given by (4.1.-15). To show this correspondence, the discrete-time version of the detection problem for OOK signal in white Gaussian is formulated below.

The signal model is

$$z(k) = n(k) \qquad m_j = 0$$

$$z(k) = n(k) + y(k) \quad : \qquad m_j = 1 \tag{4.1.-16}$$

where $n(k)$ is zero-mean white Gaussian noise with variance, V_{nn}. $y(k)$ is colored zero-mean Gaussian noise with known statistics. Then

$$\hat{z}(k|k\text{-}1,\ m_j = 0) = E\{z(k)|\underline{Z}\ (k\text{-}1),\ m_j = 0\} = E\{n(k)\} = 0$$

$$\hat{z}(k|k\text{-}1,\ m_j = 1) = E\{z(k)|\underline{Z}(k\text{-}1),\ m_j = 1\} = E\{y(k)|\underline{Z}(k\text{-}1),\ m_j = 1\} =$$

$$= \hat{y}(k|k\text{-}1) \tag{4.1.-17}$$

Also,

$$\nu(k;m_j = 0) = z(k) - \hat{z}(k|k\text{-}1,\ m_j = 0) = z(k)$$

$$\nu(k;m_j = 1) = z(k) - \hat{z}(k|k\text{-}1,\ m_j = 1) = z(k) - \hat{y}(k|k\text{-}1) \tag{4.1.-18}$$

Next,

$$V_{\nu\nu}(k|k\text{-}1,\ m_j = 0) = E\{\nu(k;\ m_j = 0)\nu^T(k;\ m_j = 0)|\underline{Z}(k\text{-}1),\ m_j = 0$$

$$= E\{z(k)z^T(k)|\underline{Z}(k\text{-}1),\ m_j = 0\}$$

$$= V_{nn}$$

$$V_{\nu\nu}(k|k\text{-}1,\ m_j = 0) = E\{\nu(k;\ m_j = 1)\nu^T(k;\ m_j = 1)|\underline{Z}(k\text{-}1),\ m_j = 1$$

(continued)

$$= E\{[z(k) - \hat{y}(k|k-1)][z(k) - \hat{y}(k|k-1)]^T | \underline{Z}(k-1), m_j = 1\} =$$

$$= E\{[\tilde{y}(k|k-1) + n(k)][\tilde{y}(k|k-1) + n(k)]^T | \underline{Z}(k-1), m_j = 1\}$$

$$= V_{\tilde{y}\tilde{y}}(k|k-1) + V_{nn} \qquad (4.1.-19)$$

where

$$\tilde{y}(k|k-1) = y(k) - \hat{y}(k|k-1) \qquad (4.1.-20)$$

The conditional densities analogous to (4.1.-13) are formed as

$$p(z(k)|\underline{Z}(k-1), m_j = 0) = \frac{1}{2\pi} [\det V_{nn}]^{-\frac{1}{2}} \exp[-\tfrac{1}{2} z^T(k) V_{nn}^{-1} z(k)]$$

$$p(z(k)|\underline{Z}(k-1), m_j = 1) = \frac{1}{2\pi} [\det (V_{nn} + V_{\tilde{y}\tilde{y}}(k|k-1)]^{-\frac{1}{2}} \cdot$$
$$\cdot \exp[-\tfrac{1}{2} (z(k) - \hat{y}(k|k-1))^T (V_{\tilde{y}\tilde{y}}(k|k-1) + V_{nn})^{-1} \cdot$$
$$\cdot (z(k) - \hat{y}(k|k-1)] \qquad (4.1.-21)$$

From (4.1.-21) the product required in (4.1.-12) yields the compound densities $p(\underline{Z}(K)|m_j = 0)$ and $p(\underline{Z}(K)|m_j = 1)$. Then the likelihood ratio is formed

$$LR \triangleq \frac{p(\underline{Z}(K)|m_j = 1)}{p(\underline{Z}(K)|m_j = 0)} =$$

$$= \left[\prod_{k=1}^{K} \frac{\det V_{nn}}{\det(V_{nn} + V_{\tilde{y}\tilde{y}}(k|k-1))}\right]^{\frac{1}{2}} \cdot \exp\left\{-\tfrac{1}{2}\left[\sum_{k=1}^{K} (z(k) - \hat{y}(k|k-1))^T \cdot \right.\right.$$
$$\left.\left. \cdot (V_{\tilde{y}\tilde{y}}(k|k-1) + V_{nn})^{-1} \cdot (z(k) - \hat{y}(k|k-1)) - z^T(k) V_{nn}^{-1} z(k)\right]\right\} \qquad (4.1.-22)$$

Since all quantities are scalar, the determinant and transpose notation may be dropped. Also (4.1.-22) is manipulated into the following form

$$LR = \left[\prod_{k=1}^{K} \frac{V_{nn}}{V_{nn} + V_{\tilde{y}\tilde{y}}(k|k-1)} \right]^{\frac{1}{2}} \cdot \exp\left[\sum_{k=1}^{K} \hat{y}(k|k-1) \cdot \right.$$

$$\cdot (V_{nn} + V_{\tilde{y}\tilde{y}}(k|k-1))^{-1} z(k) +$$

$$+ (-\tfrac{1}{2}) \sum_{k=1}^{K} \hat{y}(k|k-1)(V_{nn} + V_{yy}(k|k-1))^{-1} \hat{y}(k|k-1) +$$

$$\left. + \tfrac{1}{2} \sum_{k=1}^{K} z(k) \left[V_{nn}^{-1} - \left[(V_{nn} + V_{\tilde{y}\tilde{y}}(k|k-1))^{-1} \right] z(k) \right] \right] \quad (4.1.-23)$$

The LR of equation (4.1.-23) is similar in form to the continuous-time LR of equation (4.1.-15), with the finite summation playing the part of the integral. The dissimilarities are (i) a non-unity multiplier of the exponential term, (ii) a third term in the exponential, and (iii) a weighting which is not just the inverse of the additive noise variance (assumed unity in [33]).

It is noted that $V_{\nu\nu}(k|k-1, m_j = 1)$ is the variance of the Innovations process associated with the hypothesis, $m_j = 1$. As noted in [33], as the sampling becomes dense in the interval [0,T], to approach the continuous-time case, the white noise variance, V_{nn}, dominates the predicted error variance, $V_{\tilde{y}\tilde{y}}(k|k-1)$. Thus, in the limit for dense sampling, (4.1.-23) becomes

$$LR = \exp\left[\sum_{k=1}^{\infty} \hat{y}(k|k-1) V_{nn}^{-1} z(k) - \tfrac{1}{2} \sum_{k=1}^{\infty} \hat{y}(k|k-1) V_{nn}^{-1} \hat{y}(k|k-1) \right] \quad (4.1.-24)$$

The LR formulation for discrete-time dense sampling of (4.1.-24) is certainly a counterpart of the continuous-time LR of (4.1.-15).

(iii) Painter's Results.

Recently, Painter [29] has obtained Monte Carlo simulation results in a preliminary NASA-sponsored investigation of the optimum detector of Figures 4.1.-2 and 4.1.-3. A quaternary alphabet was used, with independent occurrence of successive symbols. The interference process consisted of colored multiplicative noise and white additive noise. No colored additive noise was implemented. The signal modulation format was an AMSK form, proposed by Aeronautical Radio, Incorporated for an industry-standard airline digital data-link [34].

In the simulation, the optimum algorithms of (4.1.-12), and (4.1.-13) were used, with the exception that $\hat{H}_o(k)$ was not included in the Kalman filter algorithms. For the first simulation runs, $\hat{\underline{\beta}}()$ was given to the Kalman filter, exactly, by hardwiring the various elements directly from the data process generator. In these runs $H_o(k)$ was set equal to the identity matrix, both in the data generator and in the Kalman filters. Thus, the first simulation runs were with perfect identification. From simulation results, measured symbol error rates were plotted. The error rate curves, so derived, displated the desired exponential decrease with decreasing white noise level. See Figure 4.1.-6.

Standard receivers, having no knowledge of the multiplicative noise, were run in parallel with the optimum interference tracking algorithms, for the purpose of comparison. The standard error rate curves displayed the expected saturation of error rate with decreasing white noise level. See Figure 4.1.-7.

The final set of Monte Carlo simulation runs employed suboptimum estimation of some of the required identification elements in $\underline{\beta}(k)$. In particular, $V_{nn}(k)$ and $\underline{\mu}(k)$, the white noise variance and channel mean value function, respectively, were estimated. During these runs the $\underline{\beta}(k)$ process was held stationary. In particular, $V_{nn}(k)$ and $\underline{\mu}(k)$ were held constant. Also, symbol timing was provided to the data processor. For some of the runs, $H_o(k)$ was held to the identity matrix. In some runs, a "phase-jitter" process, $\phi_o(k)$, based on phase-locked loop carrier phase synchronization, was inserted into $H_o(k)$. However, for these runs, $\hat{H}_o(k)$ was not inserted into the Kalman filter algorithms. The resulting inaccuracy appeared to have little effect on the resulting error rate curve. See Figure 4.1.-8.

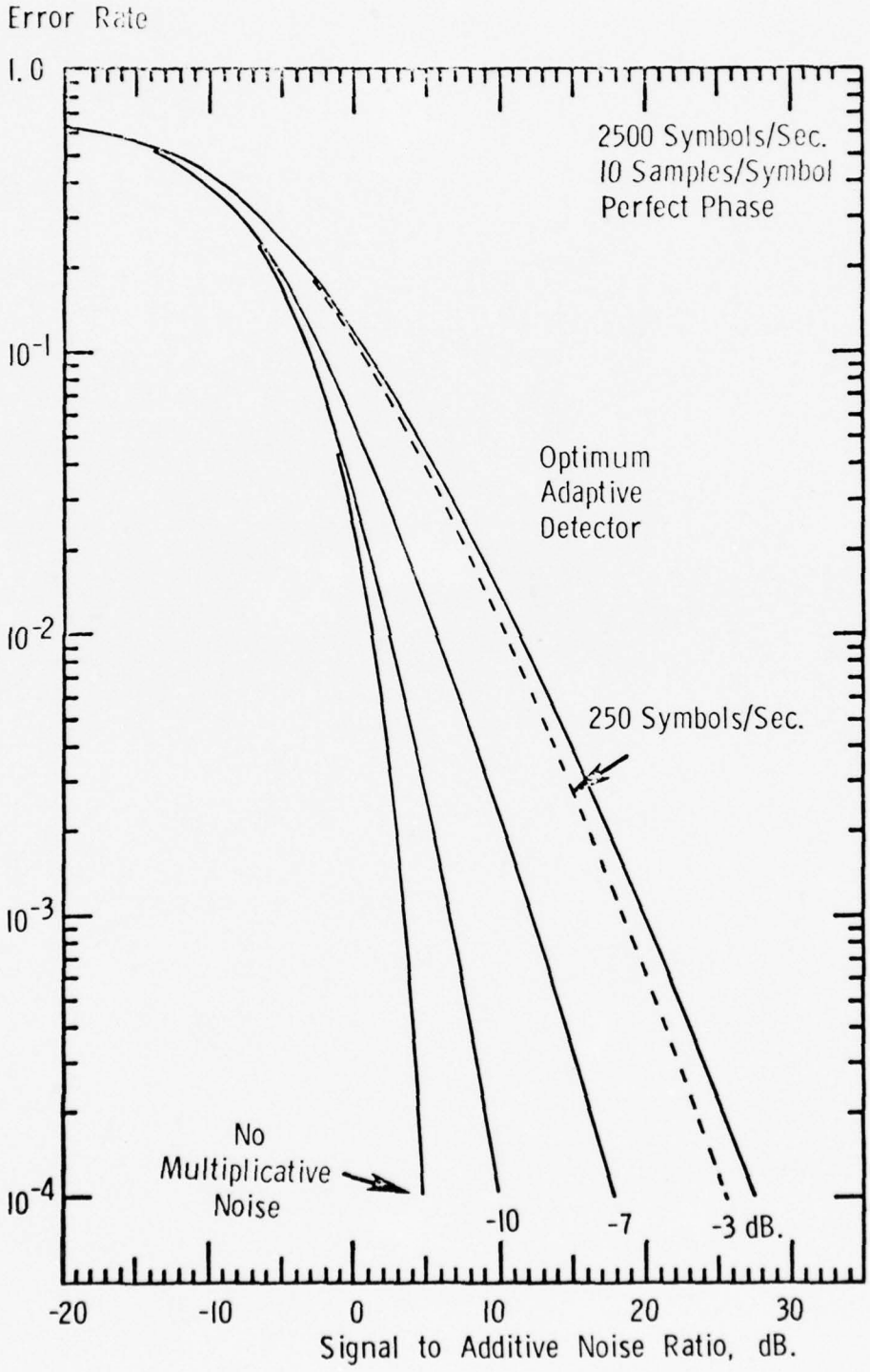


Figure 4.1.-6: Error Rate, Multiplicative Noise Only, Perfect Identification

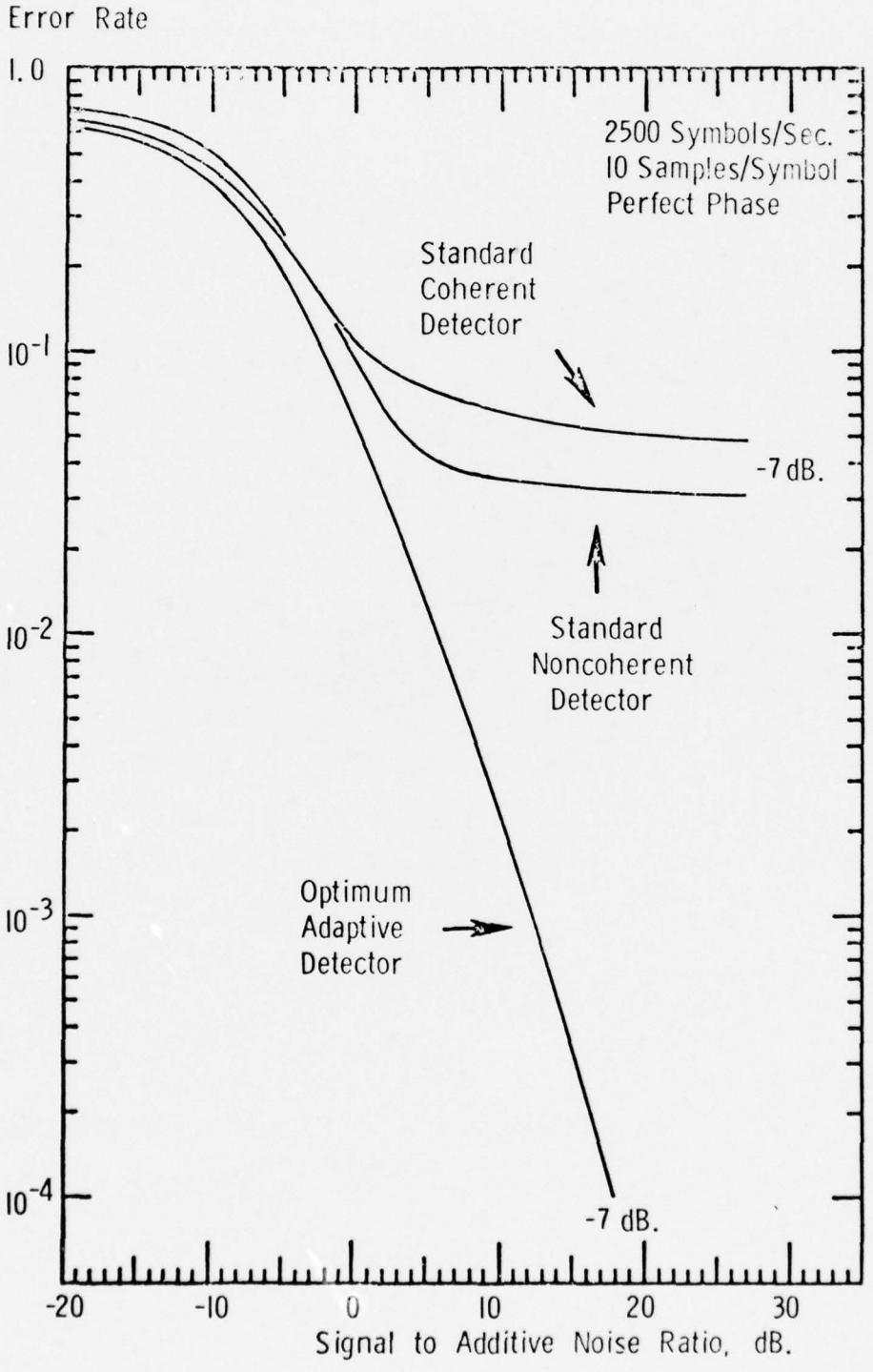


Figure 4.1.-7: Error Rate Comparison, Standard Versus Optimum.

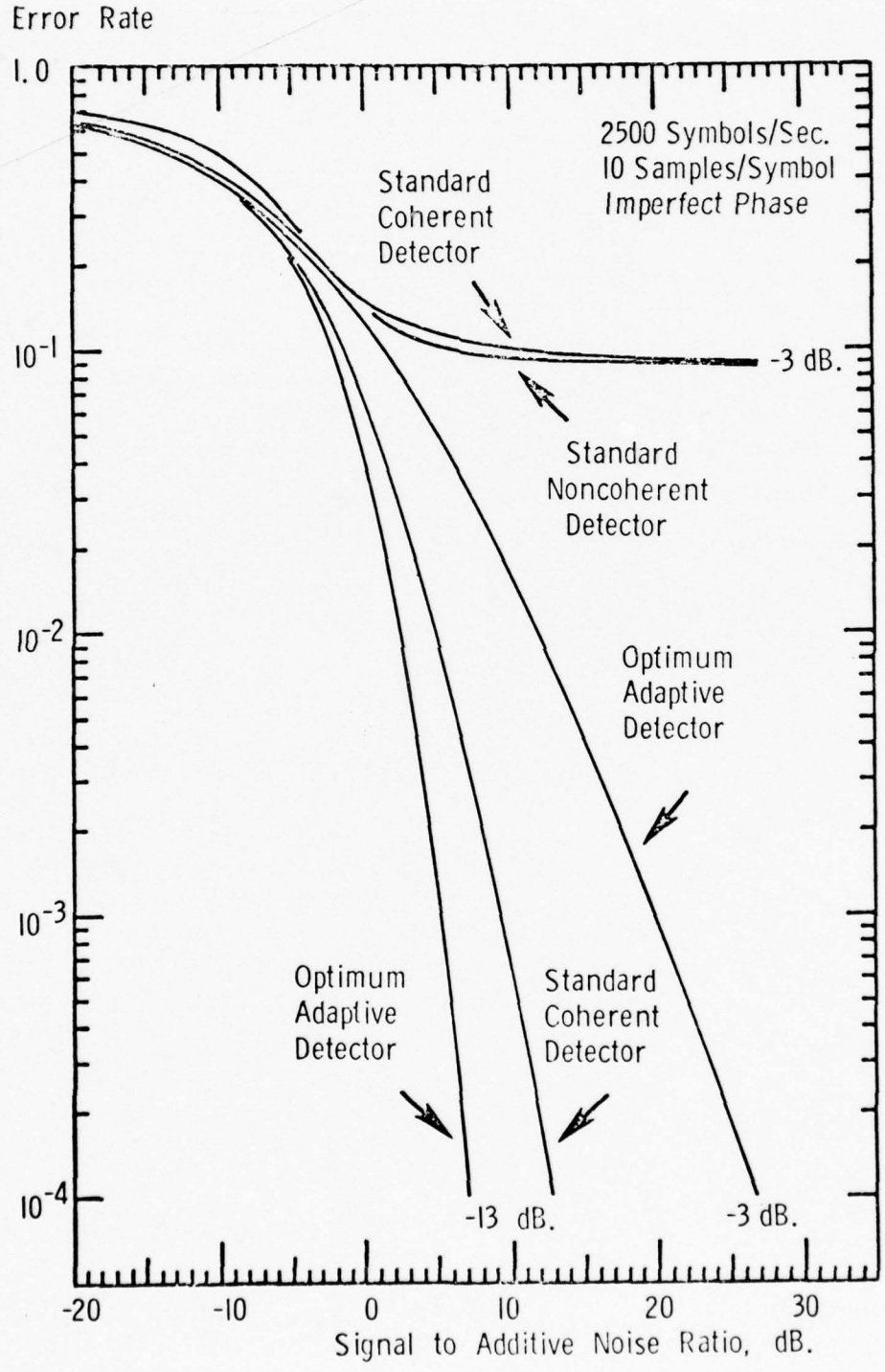


Figure 4.1.-8: Error Rate for Imperfect Phase Reference.

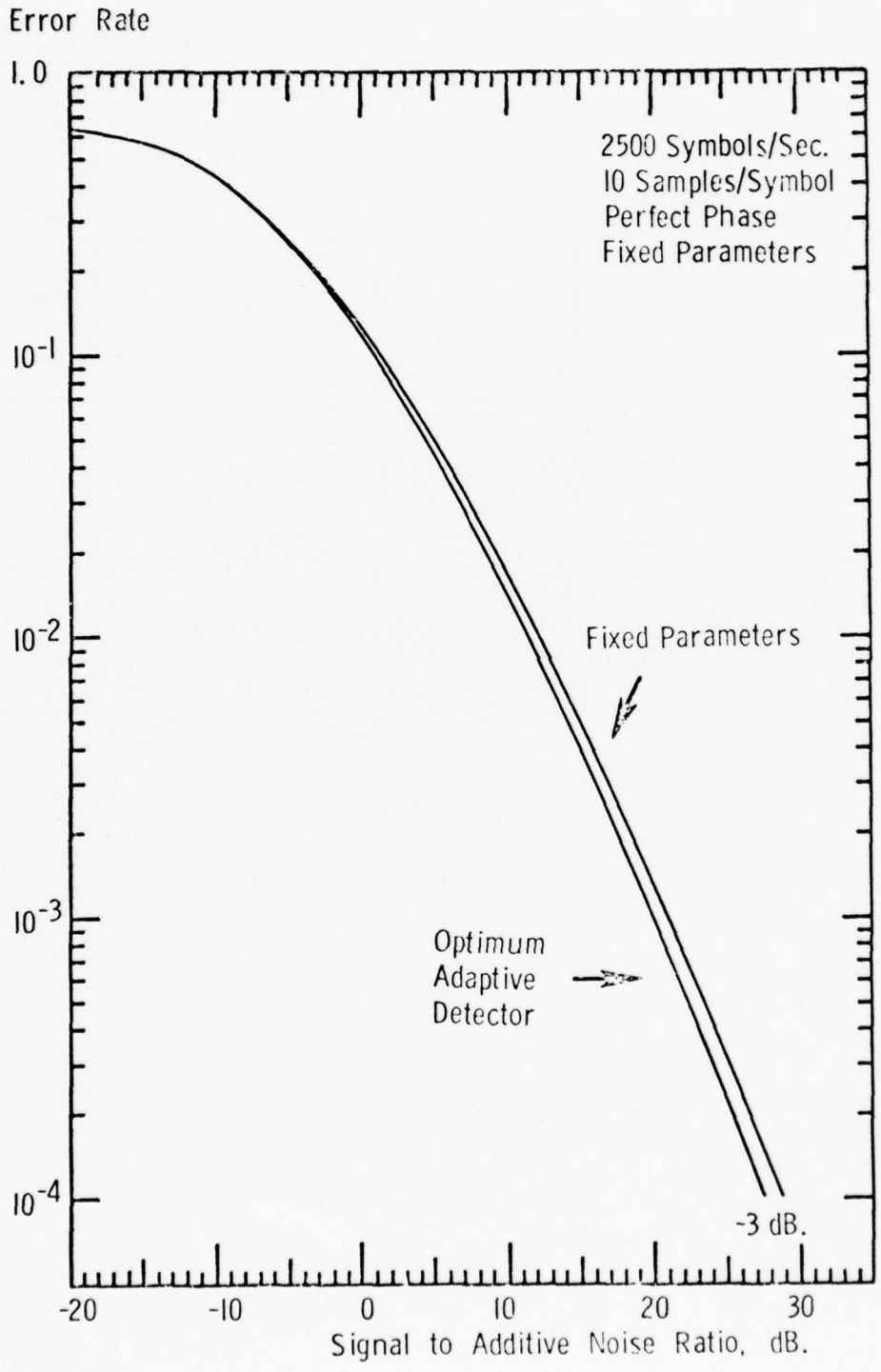


Figure 4.1.-9: Error Rate for Fixed Filter Structure.

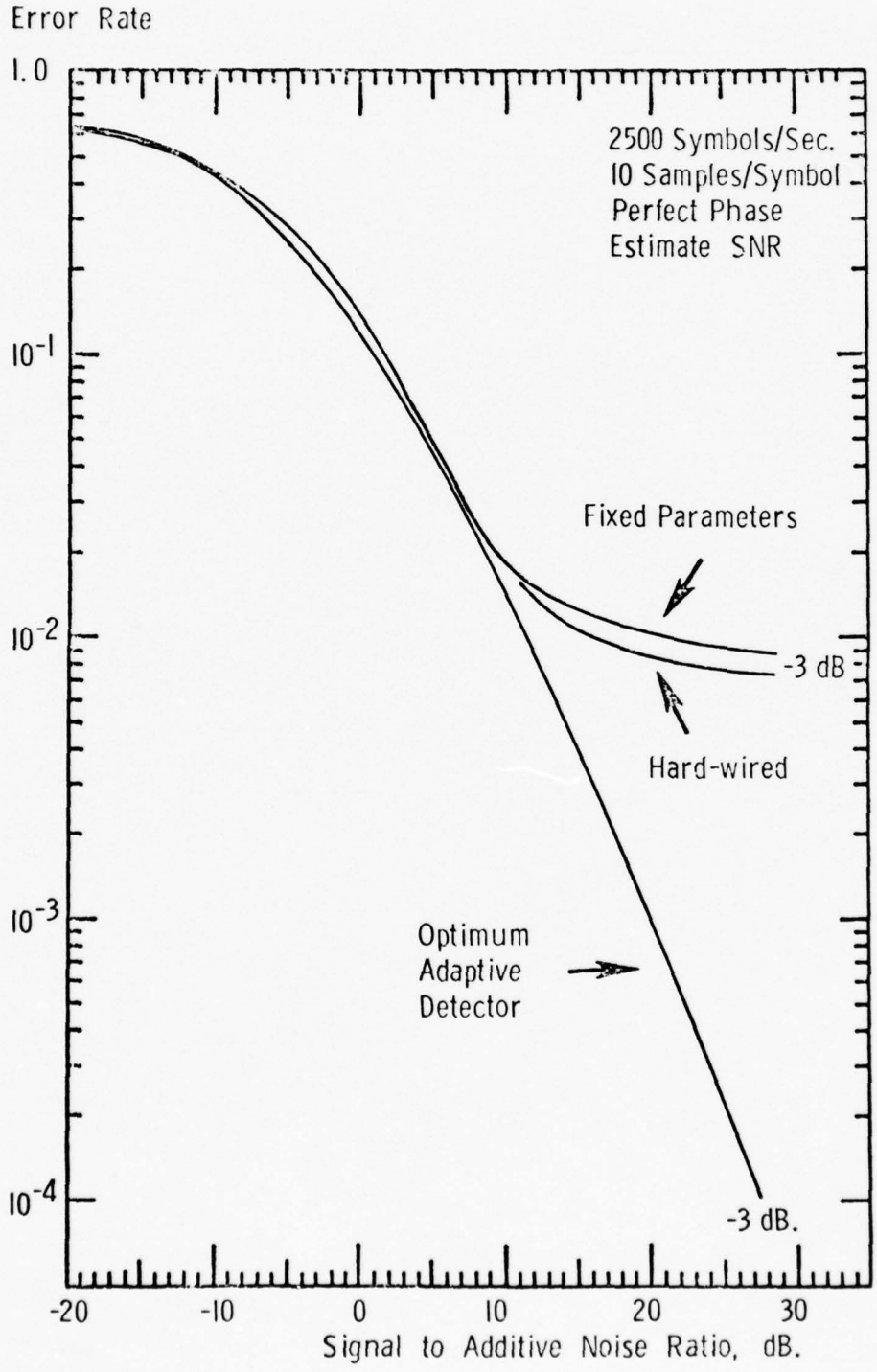


Figure 4.1.-10: Error Rate for Maximum Likelihood Identification.

In implementing the suboptimum identification estimators, it was found that the triple $\{\hat{\Gamma}(), \hat{\Phi}(), \hat{\Lambda}()\}$ could be set into the Kalman filter algorithms in a fixed configuration. All that was necessary was that $\{\hat{\Gamma}(), \hat{\Phi}(), \hat{\Lambda}()\}$ represent a filter structure which creates a process of "wider" power spectral density than that actually presented by $\underline{Y}(k)$. It is clear from work of Son and Anderson [15] that the triple $\{\hat{\Gamma}(), \hat{\Phi}(), \hat{\Lambda}()\}$ is not unique. For the optimum identification problem, the identified triple $\{\hat{\Gamma}(), \hat{\Phi}(), \hat{\Lambda}()\}$ need only be capable of recreating the autocovariance function $V_{yy}(k,j)$, for the interference process, $\underline{Y}(k)$. However, in the simulation, it was found that the measured error rate was not highly sensitive to the values of $\{\hat{\Gamma}(), \hat{\Phi}(), \hat{\Lambda}()\}$, so long as the bandwidth constraint, mentioned above, was met. See Figure 4.1.-9.

The suboptimum estimators for $\underline{\mu}(k)$ and $V_{nn}(k)$ were not conditional-mean. They were maximum-likelihood sample-mean and sample-variance estimators respectively. Moreover, they were not absolutely recursive. Since $V_{nn}(k)$ and $\underline{\mu}(k)$ were known, A Priori, to be constant, the estimates $\hat{V}_{nn}()$, $\hat{\underline{\mu}}()$, obtained by decision-direction at the end of the (J-1)st symbol interval, were used for each k in the Jth symbol interval. It was found that the suboptimum estimator for $\underline{\mu}(k)$ performed satisfactorily, in terms of the error rate measure. The error rate for the adaptive optimum detector saturated at a value lower than that for the standard detector, but at a level too high to be useful. See Figure 4.1.-10.

4.2. THE LINEAR TRACKING ALGORITHMS.

For the assumed data generating model of equation 3.6.-2 and Figure 3.6.-2, it is assumed that the rotational process which produces the unitary matrix, $H_o(k)$, is generated in the receiver itself. Thus, $H_o(k)$ is measureable and its inverse may be computed. Therefore, the first step in the detection process is to multiply the $\underline{z}(k)$ of 3.6.-2 by $H_o^{-1}(k)$. If, in practice this is not done, the prior results of [29] indicated that the effects of $H_o(k)$ are absorbed into the multiplicative noise, $\underline{y}_r(k)$, itself.

Under the assumptions concerning $H_o(k)$, the data model for use with the Kalman filter is

$$\underline{z}(k) = H_{\gamma}(k)\ \underline{\gamma}(k;m) + H_{\mu}(k;m)\underline{\mu}(k) + \underline{n}(k) \qquad (4.2.\text{-}1)$$

where the various elements of (4.2.-1) are as in (3.6.-2).

The Kalman filter corresponding to equation (4.2.-1) is depicted in Figure 4.2.-1. Because the multiplicative noise, $\underline{\gamma}_r(k)$ and the additive colored noise, $\underline{\gamma}_j(k)$, are generated independently the Kalman filter may be split into two parallel branches which are coupled only in formation of the tracking error function, the Innovations. However, because of the coupling through the Innovations, the Kalman gain equations, from which $G_r(k)$ and $G_j(k)$ are computed, are coupled.

The Kalman filter equations are

$$\hat{\underline{X}}_r(k|k\text{-}1) = \Phi_r\hat{\underline{X}}_r(k\text{-}1) \quad ; \quad \hat{\underline{X}}_r(0) = \underline{0}$$

$$\hat{\underline{X}}_j(k|k\text{-}1) = \Phi_j\hat{\underline{X}}_j(k\text{-}1) \quad ; \quad \hat{\underline{X}}_j(0) = \underline{0}$$

$$\hat{\underline{y}}_r(k|k\text{-}1) = \Lambda_r\hat{\underline{X}}_r(k|k\text{-}1)$$

$$\hat{\underline{y}}_j(k|k\text{-}1) = \Lambda_j\hat{\underline{X}}_j(k|k\text{-}1)$$

$$\hat{\underline{\gamma}}(k|k\text{-}1, m) = \left[\begin{array}{c} (H_t(k;\ m;\ \Delta)\ \hat{\underline{y}}_r(k|k\text{-}1))*h(k) \\ \hline \hat{y}_j(k|k\text{-}1) \end{array}\right]$$

$$\underline{\nu}(k) = \underline{z}(k) - H_{\gamma}(k)\ \hat{\underline{\gamma}}(k|k\text{-}1,\ m) - H_{\mu}(k;\ m)\ \underline{\mu}(k)$$

$$\hat{\underline{X}}_r(k) = \hat{\underline{X}}_r(k|k\text{-}1) + G_r(k)\ \underline{\nu}(k)$$

$$\hat{\underline{X}}_j(k) = \hat{\underline{X}}_j(k|k\text{-}1) + G_j(k)\ \underline{\nu}(k) \qquad (4.2.\text{-}2)$$

The equations of (4.2.-2) are also the Wiener filter equations when the asymptotic steady-state values are used for the Kalman gain functions, $G_r(k)$ and $G_j(k)$. Note that the filter algorithms require many of the A Priori unknown elements of the $\underline{\beta}()$ vector, such as $\underline{\mu}(k)$, Δ, etc. This shows the practical significance of the Identification problem.

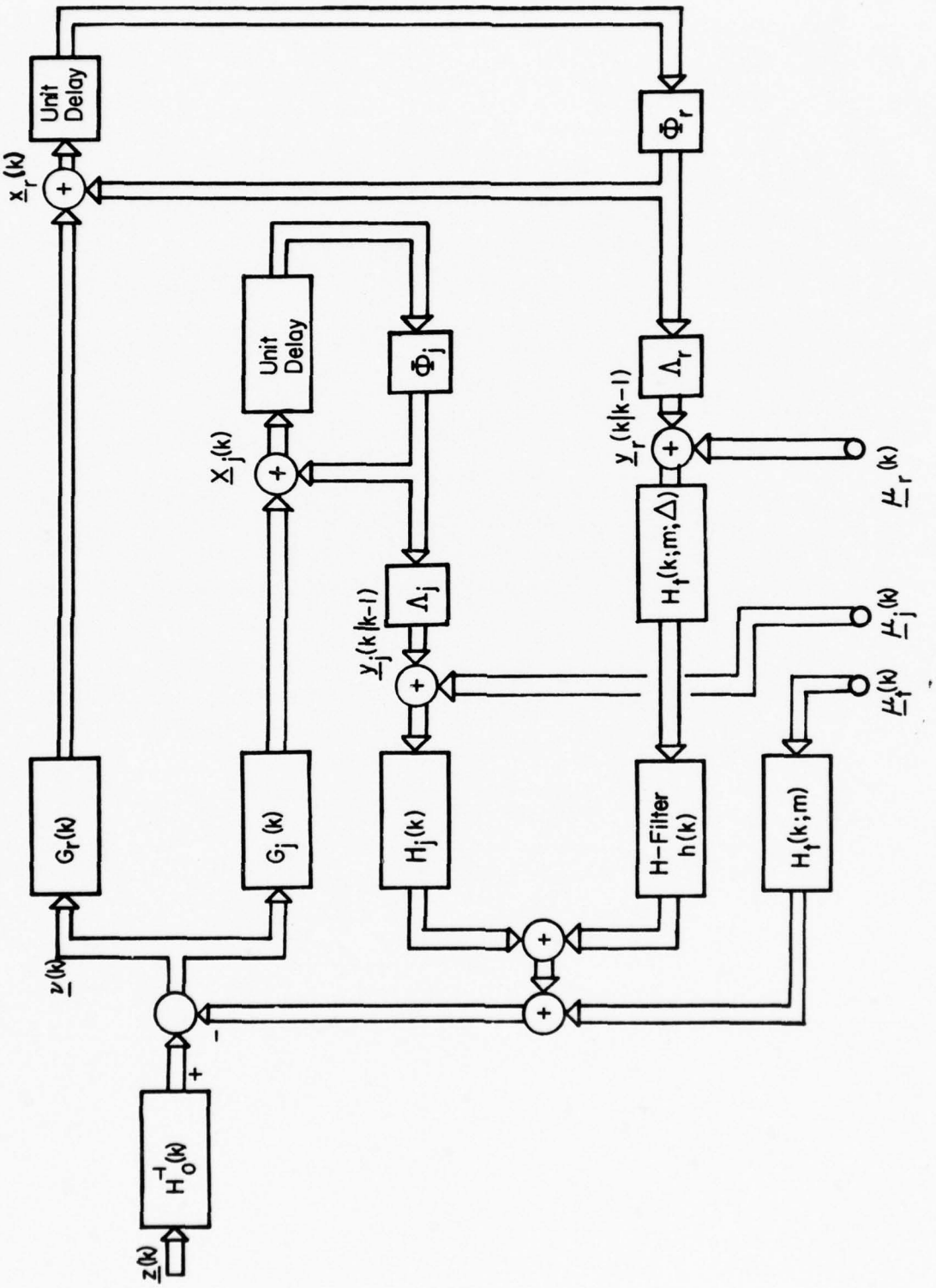


Figure 4.2.-1: Kalman Filter.

For the case of additive colored interference only, with no multiplicative noise, the Kalman gain equations for $G_j(k)$ are straight-forward. They are

$$V_{\tilde{x}_j\tilde{x}_j}(k|k-1) = \Phi_j V_{\tilde{x}_j\tilde{x}_j}(k-1)\Phi_j^T + \Gamma_j\Gamma_j^T \; : \; V_{\tilde{x}_j\tilde{x}_j}(0) = V_{x_jx_j}(0)$$

$$G_j(k) = V_{\tilde{x}_j\tilde{x}_j}(k|k-1)\Lambda_j^T H_j^T(k) \cdot [V_{nn} + H_j(k)\Lambda_j V_{\tilde{x}_j\tilde{x}_j}(k|k-1)\Lambda_j^T H_j^T(k)]^{-1}$$

$$V_{\tilde{x}_j\tilde{x}_j}(k) = [I - G_j(k) H_j(k) \Lambda_j] V_{\tilde{x}_j\tilde{x}_j}(k|k-1) \qquad (4.2.-3)$$

where V_{nn} is the 2 x 2 variance matrix for the additive white noise, $\underline{n}(k)$, as given in (3.5.-16).

The Kalman gain equations for the case of multiplicative noise only are not straight-forward for the general case when h(k) is not equal to the delta function (Kroneker delta). In place of the term of form $H\Lambda V_{\tilde{x}\tilde{x}}\Lambda^T H^T$ in equation (4.2.-3), the corresponding term in $G_r(k)$ is

$$\sum_{i=0}^{\infty}\sum_{i=0}^{\infty} H_t(k-i, m, \Delta)\Lambda_r V_{\tilde{x}_r\tilde{x}_r}(k-i, k-j)\Lambda_r^T H_T^T(k-j, m, \Delta)h(i)h(j)$$

which is essentially a double discrete convolution. It is clear from this result that the gain function, $G_r(k)$ can not be computed on line. If used at all, the Kalman filter for the delay-spread signal would require a pre-computed and stored Kalman gain function, $G_r(k)$. This is another reason for using the stationary Wiener filter in practice. For the Doppler-spread channel only, without delay-spreading, however, the gain function, $G_r(k)$, may be computed on-line, analogous to $G_j(k)$.

For the case of no delay-spreading but with both colored additive interference and colored multiplicative noise (Doppler-spreading), the Kalman gain functions may be obtained by solving the partitioned matrix equations analogous to (4.2.-3) for the compound state-vector obtained as $\underline{X}(k) = [\underline{X}_r^T(k): \underline{X}_j^T(k)]^T$.

As mentioned above, the stationary Wiener filter is obtained using the Kalman filter structure of Figure 4.2.-1 and the steady-state values for the Kalman gain functions. In general this requires solving the three gain equations for the steady-state value of the tracking error variance matrix and computing the gain functions using the steady-state error variance.

For colored additive interference or for the Doppler-spread multiplicative noise channel without delay-spreading, the gain equations have the general form

$$V(k|k-1) = \Phi V(k-1)\Phi^T + \Gamma\Gamma^T$$

$$V(k) = [I - G(k)H(k)\Lambda]V(k|k-1)$$

$$G(k) = V(k)\Lambda^T H^T(k)V_{nn}^{-1} \qquad (4.2.-4)$$

In (4.2.-4), $V(k|k-1)$ and $V(k)$ are the predicted and filtered error variance matrices, respectively. $H(k)$ is a unitary matrix representing signal modulation in the Doppler-spread case or the offset carrier effect for colored additive interference. Note that the equation for $G(k)$ is the alternate expression which is not used for recursive on-line computation.

In the steady-state, $V(k|k-1)$ and $V(k)$ are not equal but $V(k)$ and $V(k-1)$ are equal. Thus, (4.2.-4) may be manipulated to obtain

$$V(k) \triangleq V = [I - V\Lambda^T H^T(k)V_{nn}^{-1}H(k)\Lambda][\Phi V\Phi^T + \Gamma\Gamma^T] \qquad (4.2.-5)$$

where V is the constant value of the tracking error variance matrix. Equation (4.2.-5) is quadratic in V and may be solved for V. Note that a requirement for V to be constant is that $H(k)$ be unitary so that the term $H^T(k)V_{nn}^{-1}H(k)$ is constant. Given the steady-state solution for $V(k) = V$, the Wiener gain function is

$$G(k) = V\Lambda^T H^T(k)V_{nn}^{-1} \qquad (4.2.-6)$$

Note that $G(k)$ is not constant, since it contains the time-varying term, $H^T(k)$. However, the inner product $G^T(k)G(k)$, which may be interpreted as the "power" of $G(k)$ is constant.

4.3. STANDARD DETECTION ALGORITHMS.

It is desired to compare the performance of the optimum detection algorithms of (4.1.-12) with the performance of standard sampled-data detectors. For comparison purposes, binary phase-shift-keying (PSK) and frequency-shift-

keying (FSK) are chosen. The standard algorithms may be derived from consideration of Figure 4.3.-1, which shows the In-phase/Quadrature Carrier Demodulator.

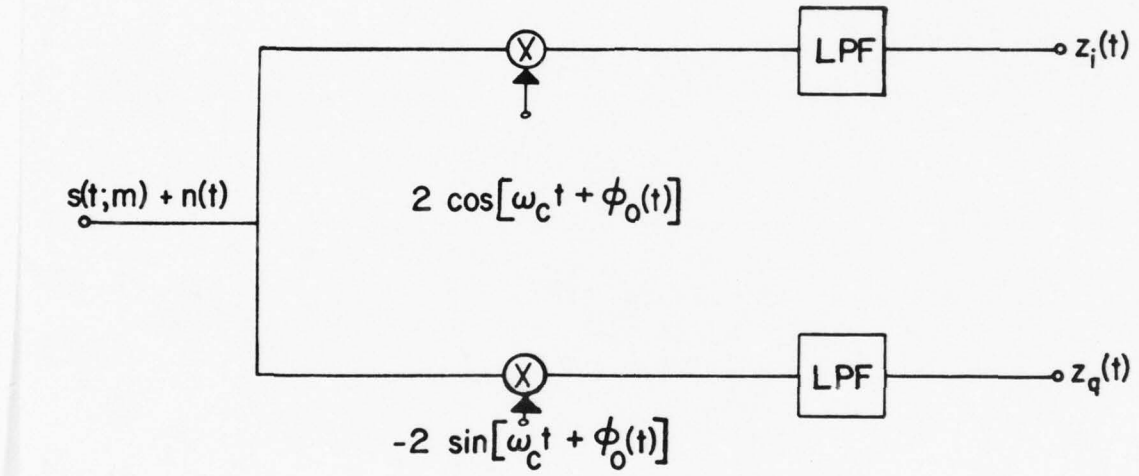


Figure 4.3.-1. I-Q Carrier Demodulator.

The continuous time input signal and noise processes are modeled as

$$s(t; m) = A(t; m)\cos[w_c t + \phi(t; m)] \quad : \quad m = 0, 1$$

$$n(t) = n_i'(t) \cos w_c t - n_q'(t) \sin w_c t \qquad (4.3.-1)$$

In (4.3.-1), $A(t; m)$ and $\phi(t; m)$ are envelope and phase functions for the modulated signal, as detailed in Section 3.2. The processes $n_i'(t)$ and $n_q'(t)$ are independent, zero-mean, white noise, with equal variances.

The output I-Q data vector is defined by

$$z(t) = \begin{bmatrix} z_i(t) \\ z_q(t) \end{bmatrix} = H_o(t)\, \underline{s}(t; m) + n(t)$$

$$H_o(t) = \begin{bmatrix} \cos\phi_o(t) & \sin\phi_o(t) \\ -\sin\phi_o(t) & \cos\phi_o(t) \end{bmatrix} \quad : \quad s(t; m) = \begin{bmatrix} A(t; m) & \cos\phi(t; m) \\ A(t; m) & \sin\phi(t; m) \end{bmatrix}$$

$$\underline{n}(t) = \begin{bmatrix} n_i(t) \\ n_q(t) \end{bmatrix} = \begin{bmatrix} n_i(t)\cos\phi_o(t) + n_q'(t)\sin\phi_o(t) \\ -n_i'(t)\sin\phi_o(t) + n_q'(t)\cos\phi_o(t) \end{bmatrix} \qquad (4.3.-2)$$

In (4.3.-2), $\phi_o(t)$ is a phase perturbation which in this model is introduced in the I-Q demodulation reference. However, in general, such a phase perturbation, which results in a rotation of the signal vector, could be introduced by the channel itself. The effect on the detection algorithms will be the same, regardless of the physical origin of $\phi_o(t)$. The good practical approximation is made that $\phi_o(t)$ is independent of $n_i'(t)$ and $n_q'(t)$. Then $n_i(t)$ and $n_q(t)$ are zero-mean and have the same variance as $n_i'(t)$ and $n_q'(t)$. $n_i(t)$ and $n_q(t)$ are taken as Gaussian, since for the given variance, a Gaussian process has maximum entropy. Hence, $\underline{n}(t)$ is Gaussian.

Sampled-data detection is accomplished by accumulating K samples of the data vector, $\underline{z}(t)$, during the symbol period of length, T. Then, the 2K-vector, $\underline{\mathcal{Z}}(K)$ is formed as

$$\underline{\mathcal{Z}}(K) = \begin{bmatrix} \underline{z}(K) \\ \underline{z}(K-1) \\ \vdots \\ \underline{z}(1) \end{bmatrix} \qquad (4.3.-3)$$

It is assumed that the symbols are equally likely. That is

$$p(m) = \tfrac{1}{2} \quad : \quad m = 0, 1 \qquad (4.3.-4)$$

Thus, the Maximum A Posteriori Probability detection strategy and the Maximum Likelihood detection strategy are identical and the symbol decision which is obtained by comparing the densities, $p(m|\underline{\mathcal{Z}}(k))$, is also obtained by comparing the densities, $p(\underline{\mathcal{Z}}(K)|m)$ for $m = 0, 1$.

In general, the required density may be obtained as the marginal density

$$p(\underline{z}(K)|m) = \int_{-\infty}^{\infty} \cdots \int p(\underline{z}(K), \underline{\phi}_0(K))\, d\underline{\phi}_0(K) \qquad (4.3.-5)$$

where the indicated formal integration is over each value of the process, $\phi_0(k)$, that is, over $\phi_0(1)$, $\phi_0(2)$, ..., $\phi_0(K)$.

4.3.1. Coherent Detection of Angle-Modulated Signals.

Coherent detection algorithms are obtained by assuming that

$$\phi_0(k) = 0 \quad \forall \quad k = 1, 2, \ldots, K \qquad (4.3.-6)$$

This is equivalent to assuming perfect phase-coherent demodulation references for the I-Q demodulator. Under this assumption, the 2K data vector is

$$\underline{z}(K) = \underline{S}(K; m) + \underline{N}(K) \qquad (4.3.-7)$$

where for angle-modulated signals,

$$S(K; m) = [\cos\phi(K; m), \sin\phi(K; m) \ldots \cos\phi(1; m), \sin\phi(1; m)]^T$$

$$\underline{N}(K) = [n_i(K), n_q(K), \ldots, n_i(1), n_q(1)]^T \qquad (4.3.-8)$$

The required density, $p(\underline{z}(K)|m)$, is Gaussian and is given by

$$p(\underline{z}(K)|m) =$$

$$= \frac{1}{(2\pi)^K (\sigma_n^2)^K} \exp\left\{-\frac{1}{2\sigma_n^2}[\underline{z}(K) - \underline{S}(K; m)]^T[\underline{z}(K) - \underline{S}(K; m)]\right\} \qquad (4.3.-9)$$

where σ_n^2 is the variance of $n_i(k)$ and $n_q(k)$. A Likelihood Ratio (LR) test is defined as

$$LR = \frac{p(\underline{z}(K)|m = 0)}{p(\underline{z}(K)|m = 1)} \underset{m=1}{\overset{m=0}{\gtrless}} 1 \qquad (4.3.-10)$$

Substituting (4.3.-9) into (4.3.-10) yields the Likelihood Ratio,

$$LR = \exp\{-\frac{1}{2\sigma_n^2}[\,-2\,\underline{z}^T(K)[\underline{S}(K;\,0) - \underline{S}(K;\,1)] +$$

$$\underline{S}^T(K;\,0)\underline{S}(K;\,0) - \underline{S}^T(K;\,1)\underline{S}(K;\,1)]\} \qquad (4.3.\text{-}11)$$

For angle-modulated signals s(t: m = 0) and s(t: m = 1) are equal power signals. Thus

$$\underline{S}^T(K;0)\underline{S}(K;0) = \underline{S}^T(K;1)\underline{S}(K;1) \qquad (4.3.\text{-}12)$$

and the Likelihood Ratio test of (4.3.-10) becomes

$$\underline{z}^T(K)\underline{S}(K;0) \underset{m=1}{\overset{m=0}{\gtrless}} \underline{z}^T(K)\underline{S}(K;1) \qquad (4.3.\text{-}13)$$

The test of (4.3.-13) is actually in the form of a correlation test, since the inner product, $\underline{z}^T(K)\underline{S}(K:\,m)$, is a measure of the correlation between the signal waveform and the data waveform.

The test of (4.3.-13) may be rewritten as

$$v = \sum_{k=1}^{K} \underline{z}^T(k)\begin{bmatrix}\cos\phi(k;0) - \cos\phi(k;1)\\ \sin\phi(k;0) - \sin\phi(k;1)\end{bmatrix} \underset{m=1}{\overset{m=0}{\gtrless}} 0 \qquad (4.3.\text{-}14)$$

where v is a "sufficient statistic" for detection.

The probability of error, P(e), for coherent detection may be obtained as

$$P(e) = P_r\{[0 < v,\ m = 1] \cup [v \leq 0,\ m = 0]\}$$

$$= \tfrac{1}{2}\left[\int_0^{\infty} p(v|m = 1)\,dv + \int_{-\infty}^{0} p(v|m = 0)dv\right] \qquad (4.3.\text{-}15)$$

where p(v|m) is the density of the sufficient statistic, conditioned on the symbol, m. The variable, v, is Gaussian, when conditioned on m,

$$v = \sum_{k=1}^{K} \{[\cos[\phi(k;0) - \phi(k;1)] - 1] + n_i(k)[\cos\phi(k;0)-\cos\phi(k;1)] + n_q(k)[\sin\phi(k;0)-\sin\phi(k;1)]\}:m = 1$$

$$v = \sum_{k=1}^{K} \{1 - \cos[\phi(k;0) - \phi(k;1)]] + n_i(k)[\cos\phi(k;0)-\cos\phi(k;1)] + n_q(k)[\sin\phi(k;0)-\sin\phi(k;1)]\}:m = 0$$

(4.3.-16)

The means and variances are

$$E\{v|0\} = -E\{v|1\} = \sum_{k=1}^{K} [1 - \cos[\phi(k;0) - \phi(k;1)]] \triangleq \mu_v$$

$$\text{var}\{v|0\} = \text{var}\{v|1\} = 2\sigma_n^2 \sum_{k=1}^{K} [1 - \cos[\phi(k;0) - \phi(k;1)]] \triangleq \sigma_v^2$$

(4.3.-17)

Then, the probability of error is

$$P(e) = \tfrac{1}{2}[1 - \text{erf}\left(\frac{\mu_v}{\sqrt{2}\,\sigma_v}\right)] \triangleq \tfrac{1}{2}[1 - \text{erf}\left[\sqrt{\frac{E(1-\rho)}{2N_o}}\right]] \qquad (4.3.-18)$$

where the implicit definition is used

$$\frac{\mu_v^2}{\sigma_v^2} = \frac{E(1-\rho)}{N_o} \qquad (4.3.-19)$$

(i) Results for PSK.

For PSK, the envelope and phase functions for the modulated signal are

$$A(k;0) = A(k;1) = 1 \quad ; \Delta\phi \text{ is phase deviations in radians}$$

$$\phi(k;0) = -\phi(k;1) = \Delta\phi \qquad (4.3.-20)$$

Then

$$\frac{\mu_v^2}{\sigma_v^2} = \frac{K[1 - \cos(2\Delta\phi)]^2}{2\sigma_n^2 K[1 - \cos(2\Delta\phi)]} = \frac{\sin^2(\Delta\phi)}{\sigma_n^2/K} \qquad (4.3.-21)$$

The term, ρ, is correlation coefficient between the two transmitted signals. For PSK,

$$\rho = -1 \qquad (4.3.-22)$$

Thus, for PSK

$$\frac{E}{N_o} = \frac{\frac{1}{2}\sin^2(\Delta\phi)}{\sigma_n^2/K} \qquad (4.3.-23)$$

(ii) Results for FSK.

For FSK, the envelope and phase functions for the modulated signal are

$$A(k;0) = A(k;1) = 1$$

$$\phi(k;0) = -\phi(k;1) = \Delta\omega \cdot t_k$$

; $\Delta\omega$ is frequency deviation in radians/second
T is symbol period

$$t_k = (k - \tfrac{1}{2})\frac{T}{K} \qquad (4.3.-24)$$

Now, define the FSK tone frequency, ω_T, as

$$\omega_T = \frac{2\pi}{T} \qquad (4.3.-25)$$

Then, the equivalent phase deviation for FSK is

$$\Delta\phi = \frac{\Delta\omega}{\omega_T} \qquad (4.3.-26)$$

and

$$\Delta\omega \cdot t_k = \Delta\phi \cdot \omega_T \cdot (k - \tfrac{1}{2})\frac{T}{K} = \frac{2\pi}{K}(k - \tfrac{1}{2})\Delta\phi \qquad (4.3.-27)$$

Then,

$$\frac{\mu_v^2}{\sigma_v^2} = \frac{\sum_{k=1}^{K} 2\sin^2(\Delta\omega \cdot t_k)]^2}{2\sigma^2 \sum_{k=1}^{K} 2\sin^2(\Delta\omega \cdot t_k)} = \frac{\frac{1}{K}\sum_{k=1}^{K} \sin^2(\Delta\omega \cdot t_k)}{\sigma_n^2/K} =$$

$$= \frac{\frac{1}{K}\sum_{k=1}^{K} \sin^2[\frac{2\pi}{K}(k - \tfrac{1}{2})\Delta\phi]}{\sigma_n^2/K} \qquad (4.3.-28)$$

Provided that the equivalent phase deviation (or frequency deviation is chosen so that $\rho \simeq 0$, which is standard practice in the non-coherent case.

$$\frac{E}{N_o} = \frac{\frac{1}{K}\sum_{k=1}^{K} \sin^2(\frac{2\pi}{K}(k - \tfrac{1}{2})\Delta\phi)}{\sigma_n^2/K} = \frac{1 - \frac{1}{K}\sum_{k=1}^{K} \cos^2(\frac{2\pi}{K}(k - \tfrac{1}{2})\Delta\phi)}{\sigma_n^2/K} \qquad (4.3.-29)$$

(iii) <u>The Recursive Coherent Detection Algorithm</u>.

Since generally it is preferable to have the detection processing recursive, the Likelihood Ratio test of (4.3.-13) is now converted to recursive form. From (4.3.-14), the recursive Likelihood Ratio test may be written as

$$LR(0) = 0$$

$$LR(k) = \underline{z}^T(k)\,\underline{r}(k) + LR(k - 1) : \quad k = 1, 2, \ldots, K$$

(continued)

$$\underline{r}(k) = \begin{bmatrix} \cos\phi(k;0) - \cos\phi(k;1) \\ \sin\phi(k;0) - \sin\phi(k;1) \end{bmatrix} = \begin{bmatrix} 0 \\ 2\sin(\Delta\phi) \end{bmatrix} \quad ; \quad \text{PSK}$$

$$= \begin{bmatrix} 0 \\ 2\sin(\Delta\omega t_k) \end{bmatrix} \quad ; \quad \text{FSK}$$

$$LR(K) \underset{m=1}{\overset{m=0}{\gtrless}} 0 \qquad (4.3.-30)$$

Note that the reference signal, $\underline{r}(k)$, contains zero as the in-phase channel reference. This means that the standard detector uses only the quadrature channel signal component to make the symbol decision. This is not the case for the optimum detector for multiplicative noise (Doppler-spread channel). There, generally, a component of the signal is rotated into the in-phase channel by the multiplicative noise. Note that the same comment would apply if the I-Q carrier demodulator phase reference, $\phi_o(k)$, were non-zero.

4.3.2. Non-Coherent Detection of Angle-Modulated Signals.

Non-coherent detection algorithms are obtained by assuming that

$$\phi_o(k) = \phi_o \quad \forall\ k = 1, 2, \ldots, K \qquad (4.3.-31)$$

where ϕ_o is a random variable which is uniformly distributed over the interval, $[0, 2\pi]$. This is equivalent to assuming that the reference phase for the coherent I-Q demodulator is completely unknown, but is constant over the symbol period. Then, the required density is obtained as

$$p(\underline{z}(K)|m) = \frac{1}{2\pi} \int_0^{2\pi} p(\underline{z}(K)|m, \phi_o)d\phi_o \qquad (4.3.-32)$$

where $\underline{z}(K)$ is defined by (4.3.-3) and

$$\underline{z}(k) = H(\phi_o)\underline{s}(k;m) + \underline{n}(k)$$

$$H(\phi_o) = \begin{bmatrix} \cos\phi_o & \sin\phi_o \\ -\sin\phi_o & \cos\phi_o \end{bmatrix} \quad ; \quad \underline{s}(k;m) = \begin{bmatrix} \cos\phi(k;m) \\ \sin\phi(k;m) \end{bmatrix}$$

$$\underline{n}(k) = \begin{bmatrix} n_i(k) \\ n_q(k) \end{bmatrix} \quad ; \quad k = 1, 2, \ldots, K \qquad (4.3.-33)$$

Now,

$$p(\underline{z}(K)|m,\phi_o) =$$

$$= \frac{1}{(2\pi)^K(\sigma_n^2)^K} \exp\left\{-\frac{1}{2\sigma_n^2}[\underline{z}(K) - \overline{\underline{z}}(K|m,\phi_o)]^T[\underline{z}(K) - \overline{\underline{z}}(K|m,\phi_o)]\right\}$$

$$= \frac{1}{(2\pi)^K(\sigma_n^2)^K} \exp\left\{-\frac{1}{2\sigma_n^2}\sum_{k=1}^{K}[\underline{z}(k) - \overline{\underline{z}}(k|m,\phi_o)]^T[\underline{z}(k) - \overline{\underline{z}}(k|m,\phi_o)]\right\}$$

(4.3.-34)

where

$$\overline{\underline{z}}(k|m,\phi_o) = E\{\underline{z}(k)|m,\phi_o\}$$

$$= H(\phi_o)\underline{s}(k;m) \qquad (4.3.-35)$$

Then

$$p(\underline{z}(K)|m,\phi_o) = \frac{1}{(2\pi)^K(\sigma_n^2)^K} \exp\left\{-\frac{1}{2\sigma_n^2}\sum_{k=1}^{K}[\underline{z}^T(k)\underline{z}(k)\right.$$

$$- 2\,\underline{z}^T(k)H(\phi_o)\underline{s}(k;m)$$

$$\left. + \underline{s}^T(k;m)H^T(\phi_o)H(\phi_o)\underline{s}(k;m)]\right\}$$

$$= Q(K;m)\exp\left[\frac{1}{\sigma_n^2}\sum_{k=1}^{K}\underline{z}^T(k)H(\phi_o)\underline{s}(k;m)\right]$$

(continued)

$$Q(K;m) = \frac{1}{(2\pi)^K(\sigma_n^2)^K} \exp\left\{ -\frac{1}{2\sigma_n^2} \sum_{k=1}^{K} [\underline{z}^T(k)\underline{z}(k) + \underline{s}^T(k;m)\underline{s}(k;m)] \right\} \tag{4.3.-36}$$

Next, (4.3.-36) is manipulated into the form

$$p(\underline{z}(K)|m,\phi_o) = Q(K;m) \exp[a(K;m) \cos\phi_o + b(K;m) \sin\phi_o]$$

$$a(K;m) = \frac{1}{\sigma_n^2} \sum_{k=1}^{K} \underline{z}^T(k) \begin{bmatrix} \cos\phi(k;m) \\ \sin\phi(k;m) \end{bmatrix}$$

$$b(K;m) = \frac{1}{\sigma_n^2} \sum_{k=1}^{K} \underline{z}^T(k) \begin{bmatrix} \sin\phi(k;m) \\ -\cos\phi(k;m) \end{bmatrix} \tag{4.3.-37}$$

Finally the integral of (4.3.-32) may be performed to obtain

$$p(\underline{z}(K)|m) = \frac{1}{2\pi} \int_0^{2\pi} Q(K;m) \exp[a(K;m) \cos\phi_o + b(K;m) \sin\phi_o] d\phi_o$$

$$= Q(K;m)\, I_o[\sqrt{a^2(K;m) + b^2(K;m)}] \tag{4.3.-38}$$

where $I_o()$ is the Modified Bessel Function.

Because $\underline{s}(k;0)$ and $\underline{s}(k;1)$ are equal power signals

$$Q(K;0) = Q(K;1) \tag{4.3.-39}$$

Because $I_o()$ is monotonically increasing for positive increasing agrument, and because of (4.3.-39), the Likelihood Ratio test for non-coherent detection of angle-modulated signals becomes

$$a^2(K;0) + b^2(K;0) \underset{m=1}{\overset{m=0}{\gtrless}} a^2(K;1) + b^2(K;1) \tag{4.3.40}$$

(i) Recursive Detection of FSK.

For FSK, $\phi(k;m)$ are as given in (4.3.-24) for m = 0, 1. The

recursive algorithm is defined by

$$a(k;0) = \underline{z}^T(k)\begin{bmatrix}\cos(\Delta\omega t_k)\\ \sin(\Delta\omega t_k)\end{bmatrix} + a(k-1;\,0)$$

$$b(k;0) = \underline{z}^T(k)\begin{bmatrix}\sin(\Delta\omega t_k)\\ -\cos(\Delta\omega t_k)\end{bmatrix} + b(k-1;\,0)$$

$$a(k;1) = \underline{z}^T(k)\begin{bmatrix}\cos(\Delta\omega t_k)\\ -\sin(\Delta\omega t_k)\end{bmatrix} + a(k-1;\,1)$$

$$b(k;1) = \underline{z}^T(k)\begin{bmatrix}-\sin(\Delta\omega t_k)\\ -\cos(\Delta\omega t_k)\end{bmatrix} + b(k-1;\,1)$$

$$a(0;0) = b(0;0) = a(0;1) = b(0;1) = 0$$

$$k = 1, 2, \ldots, K$$

$$LR(k) = a^2(k;0) - a^2(k;1) + b^2(k;0) - b^2(k;1)$$

$$LR(K) \underset{m=1}{\overset{m=0}{\gtrless}} 0 \qquad (4.3.-41)$$

4.4. IMPLICATIONS ON SIGNAL DESIGN AND CODING.

The Integrated Detection, Estimation and Identification (IDEI) algorithms represent a new and different approach to Maximum A Posteriori Probability detection of M-ary signals. The algorithms provide "gain" against "Doppler-spreading colored multiplicative noise and colored additive processes by tracking the interference waveforms, themselves. Since the detection gain depends on tracking the interference (in the presence of receiver generated additive white noise), the narrower the interference spectrum is, the better the tracking and detection performance will be. This observation is based on the assumption of fixed white noise spectral density, fixed colored interference power and fixed sampling rate. In the case of intentional colored interference, game theoretic considerations imply that the interference spectrum width should be commensurate with the signal spectrum width. This and the tracking requirement imply that spread spectrum signalling may not be

the best choice, A Priori, for use with IDEI reception. Thus, the question is, "How can Detection, Estimation, Identification, and Signal Design and Coding be integrated in a rigorous yet natural manner? The answer lies in the area of "Probabilistic Decoding" [35], as opposed to "Algebraic Decoding" [36].

The distinction between algebraic and probabilistic decoding is described below, with reference to Figure 4.4.-1. The coder is a deterministic mapping of sequences of source symbols, $m_j^* \in D_Q$, to sequences of modulator input symbols, $m_j \in D_M$, where D_Q and D_M are Q-ary and M-ary alphabet sets, respectively. The modulator is a memoryless device which, for each symbol, m_j, of the input sequence, produces at its output a waveform, $y_t(t; m_j)$, on the time interval, $[(j-1)T_s, j\,T_s]$. Data transmission is accomplished by transmitting into the channel a sequence of waveforms corresponding to the modulator input sequence. Let the j^{th} N-sequence of modulator input symbols be denoted

$$w_j^N = \{m_{ji}\}_{i=1}^N \tag{4.4.-1}$$

Then the corresponding modulator output waveform is

$$y_t(t; w_j^N) = \{y_t(t; m_{ji});\ ((i-1) + (j-1)N)T_s \leq t < (i+(j-1)N)T_s\}_{i=1}^N \tag{4.4.-2}$$

The compound modulator output waveform of (4.4.-2), on passing through the channel, produces a compound received waveform, $\zeta(t; w_j^N)$, which may be viewed as a sequence of waveforms as

$$\zeta(t; w_j^N) = \{\zeta(t; m_{ji})\ ;$$

$$((i-1) + (j-1)N)T_s \leq t < (i+(j-1)N)T_s\}_{i=1}^N \tag{4.4.-3}$$

The $\zeta(t; m_{ji})$ are produced in accordance with the probabilistic mapping implied by the particular channel model being used.

The demodulator processes the individual members, $\zeta(t; m_{ji})$, of the received waveform sequence, to produce a sequence of numbers, $\{d_i\}_{i=1}^{nN}$, $d_i \in D$. The specification of n, N, and D depends on the particular decoding

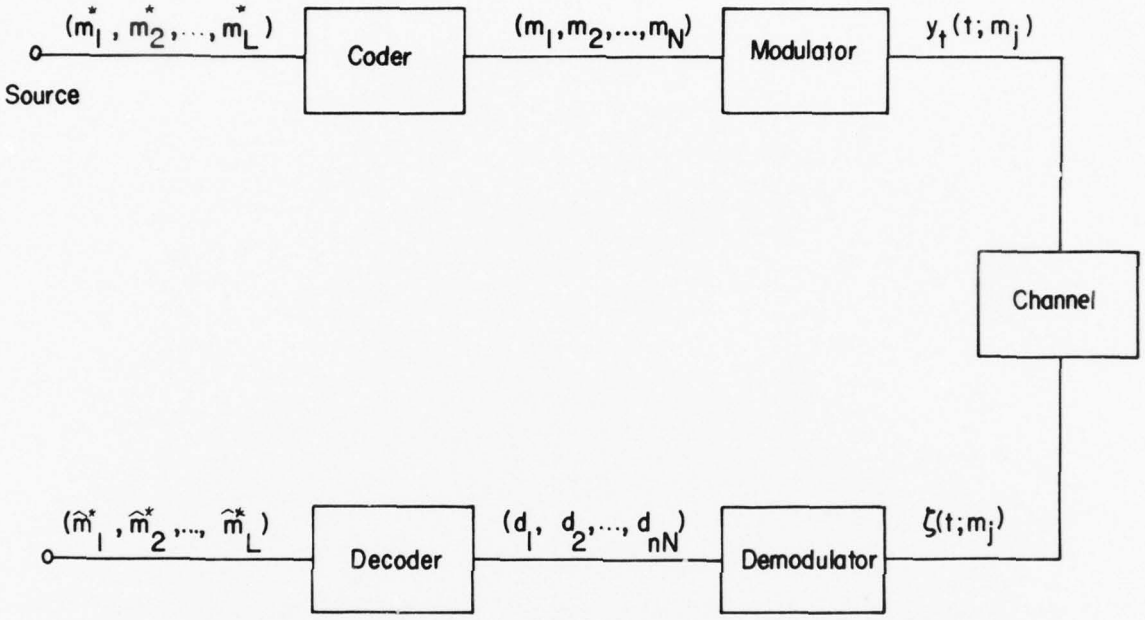


Figure 4.4.-1: Communication System Block Diagram.

strategy employed, and is made clear, below. The decoder processes the sequence, $\{d_i\}_{i=1}^{nN}$, to produce decisions, or estimates $\hat{m}_j^*$ on the corresponding source symbols, m_j^*.

In algebraic decoding, the sequence, $\{d_i\}_{i=1}^{nN}$, produced at the demodulator output, is an N-sequence of estimates, $\{\hat{m}_{ji}\}_{i=1}^{N}$, of an N-sequence of modulator input symbols, $\{m_{ji}\}_{i=1}^{N}$. In this case, $D = D_M$ and n=1. In the usual binary case, $D = \{0, 1\}$, and the d_i are binary symbols. An algebraic decoder attempts to exploit redundancy introduced in the coder my mapping the demodulator output sequence $\{d_i\}_{i=1}^{N}$, into an admissible coder input sequence, $\{\hat{m}_{ji}^*\}_{i=1}^{N}$.

Massey [37] illuminates some of the limitations that algebraic decoding places on overall communication system performance. Intuitively, if the de-

modulator produces only a decision estimate, $\hat{m}_{ji}$, of the modulator input, m_{ji}, then any available information on the quality of the decision is not utilized. Thus, the algebraic decoder must necessarily treat each of the elements of a sequence of demodulated symbols, $\{\hat{m}_{ji}\}_{i=1}^{N}$, as being of equal quality. This loss of information, due to the demodulator's implementation of a "hard decision" strategy, limits the performance of the total system formed of the coder/modulator/demodulator/decoder. For the case of transmission through a discrete, memoryless channel, such as an additive white Gaussian noise channel, Massey [37] shows that the overall system performance is improved by probabilistic decoding using a demodulator having an output alphabet larger than the modulator input alphabet. The limiting case of an infinite output alphabet results when the demodulator output consists of a vector of outputs from a set of filters matched to the channel waveform set, $\{\zeta(t; m_{ji}); i=1,N\}$.

Chase [38] discusses the use of channel quality information in probabilistic decoding of block codes for channels with memory due to correlated fading (such as the aeronautical data-link example). Block encoding of data produces binary word sequences, $\{m_{ji}\}_{i=1}^{N}$, which phase-shift-key a set of N orthogonal FDM tones in a Kineplex [39]-type modem. In addition to binary decisions, $\{\hat{m}_{ji}\}_{i=1}^{N}$, the demodulator provides the decoder with a reliability measure sequence, $\{\alpha_{ji}\}_{i=1}^{N}$, which allows the decoder to weight certain demodulator symbol decisions more heavily than others in the decoding process. Chase [38] shows that this probabilistic decoding provides a significant performance improvement for both simulated and physical channels. Unfortunately, Chase's system exhibits a saturation of the error rate curve for large signal-to-noise ratios, which he calls, an "irreducible" error rate due to time and frequency-selective fading. Such saturation was also reported in [29] for demodulators which did not properly process the multiplicative noise (which causes time-selective fading). A key question is to what extent the IDEI algorithms are subject to error rate saturation and to what extent such saturation can be reduced using probabilistic decoding.

The IDEI algorithms derived above as equation (4.1.-12), provide a rigorous, yet natural, means for integrating the function of single-symbol probabilistic decoding with those of detection, estimation and identification. Block coding and convolutional coding are treated below.

4.4.1. Block Coding.

Suppose that the coder uses block encoding, mapping a source symbol sequence of length L into a modulator input symbol sequence of length N. Since there are Q symbols in the source alphabet, it follows that there are Q^L possible sequences, each, for the source and for the modulator input. Let the j^{th} modulator input sequence, w_j, be as in (4.4.1). Let the resulting channel output waveform, $\zeta(t; w_j)$, be reduced to a discrete-time sampled-data vector, $\underline{z}_j(k)$ by sampling at a fixed uniform rate of K samples per symbol, m_{ji}, as per (3.5.-6). The vector, $\underline{z}_j(k)$, then contains KN sample vectors, $\underline{z}(k)$. Now, define a vector of code words, $\underline{w}_j(k)$, as

$$\underline{w}_j(k) = \left[\underline{w}_j^N(k), \ldots, \underline{w}_2^N(k), \underline{w}_1^N(k)\right]^T = \begin{bmatrix} \underline{w}_j^N(k) \\ \hline \underline{w}_{j-1}(k) \end{bmatrix} \qquad (4.4.-4)$$

where the elements of $\underline{w}_j()$ are vector representations of the block sequences of symbols. That is

$$\underline{w}_j^N(k) = \left[m_{jN}(k), \ldots, m_{j2}(k), m_{j1}(k)\right]^T = \begin{bmatrix} m_{jN}(k) \\ \hline \underline{w}_j^{N-1}(k) \end{bmatrix} \qquad (4.4.-5)$$

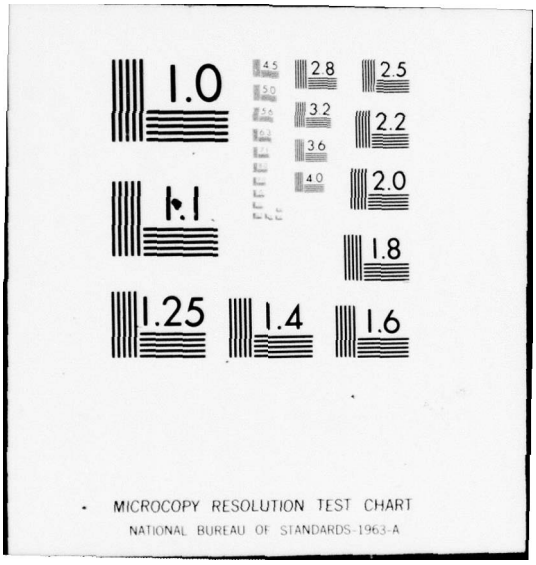
and the $m_{ji}(k)$ are the M-ary symbols in the j^{th} modulator input sequence.

Corresponding to $\underline{w}_j(k)$ is the total data vector, $\underline{Z}_j(k)$, containing jNK sample vectors, $\underline{z}(k)$, or j "block data vectors" $\underline{Z}_i^N(k)$ for i = 1, 2, ..., j. MAP detection of the entire vector of code words is performed by computing a statistic, S, proportional to the A Posteriori probability $p(\underline{w}_j(k)|\underline{Z}_j(k))$.

The same development of the present statistic, S, may be followed as was previously for symbol by symbol detection. Only now, major decisions are made at the end of each code word, $\underline{w}_i^N(k)$. In particular, assume that at the $(j-1)^{st}$ decision state, the probability, $p(\underline{w}_{j-1}(k)|\underline{Z}_{j-1}(k))$ is maximized by $\underline{w}_{j-1}(k) = \hat{\underline{w}}_{j-1}(k)$. Also assume at that stage that a very good conditional-mean estimate of the identification parameter vector, $\underline{B}_{j-1}(k)$, is available as $\hat{\underline{B}}_{j-1}(k)$. Then, the statistic, S, is decision-directed on the $(j-1)^{st}$ code word vector, and becomes

AD-A048 181 TEXAS A AND M UNIV COLLEGE STATION DEPT OF ELECTRICA--ETC F/G 17/4
LOW COST ANTI-JAM DIGITAL DATA-LINKS TECHNIQUES INVESTIGATIONS.(U)
JUN 77 J H PAINTER, J N HOLYOAK, C J YOON F33615-75-C-1011
UNCLASSIFIED AFAL-TR-77-104 NL

2 of 3
ADA048 181


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

$$S = \int \ldots \int \; p(\underline{Z}_j^N(k),\ \underline{B}_j^N(k),\ \underline{W}_j^N(k)\ \underline{z}_{j-1},\ \hat{\underline{w}}_{j-1},\ \hat{\underline{B}}_{j-1}) d\ \underline{B}_j^N(k) \tag{4.4.-6}$$

where, now, $\underline{W}_j^N(k)$ is given by (4.4.-5) and the $\underline{Z}_j^N(k)$, $\underline{B}_j^N$ are "block vectors" defined by

$$\underline{Z}_j^N(k) = \begin{bmatrix} \underline{Z}_{jN}(k) \\ \hline \underline{Z}_j^{N-1}(k) \end{bmatrix} \quad , \quad \underline{B}_j^N(k) = \begin{bmatrix} \underline{B}_{jN}(k) \\ \hline \underline{B}_j^{N-1}(k) \end{bmatrix} \tag{4.4.-7}$$

and the $\underline{Z}_{ji}(k)$, $\underline{B}_{ji}(k)$ each contain K sample vectors $\underline{z}(k)$, $\underline{\beta}(k)$, respectively. Now, by straight-forward repeated application of Bayes rule, (4.4.-6) is transformed to

$$S = \int \ldots \int \prod_{i=1}^{N} p(\underline{Z}_{ji}(k), \underline{B}_{ji}(k) \Big| m_{ji}(k), \underline{Z}_j^{i-1}(k), \underline{B}_j^{i-1}(k), \underline{W}_j^{i-1}(k), \cdot$$

$$\cdot\ \underline{z}_{j-1}, \hat{\underline{w}}_{j-1}, \hat{\underline{B}}_{j-1})\ p(m_{ji}(k) \Big| \underline{W}_j^{i-1}(k), \underline{w}_{j-1})\ d\ \underline{B}_j^N(k) \tag{4.4.-8}$$

which is equivalent to (4.1.-8), above. Equation (4.4.-8) is further transformed as

$$S = \int \ldots \int \prod_{i=1}^{N} p(\underline{Z}_{ji}(k) | \underline{B}_{ji}(k), m_{ji}(k), \underline{Z}_j^{i-1}(k), \underline{B}_j^{i-1}(k), w_j^{i-1}(k), \cdot$$

$$\cdot\ \underline{z}_{j-1}, \hat{\underline{w}}_{j-1}, \hat{\underline{B}}_{j-1}) \cdot p(\underline{B}_{ji}(k) | m_{ji}(k), \underline{Z}_j^{i-1}(k), \underline{B}_j^{i-1}(k), \underline{w}_j^{i-1}(k) \cdot$$

$$\cdot\ \underline{z}_{j-1}, \hat{\underline{w}}_{j-1}, \hat{\underline{B}}_{j-1}) \cdot p(m_{ji}(k) | \underline{W}_j^{i-1}(k), \hat{\underline{w}}_{j-1})]\ d\ \underline{B}_j^N(k) \tag{4.4.-9}$$

Equation (4.4.-9) implies an averaging type algorithm consisting of Q^L parallel branches, one for each unique modulator input code word.

To minimize demodulator complexity, it is assumed that at the end of the $(i-1)^{st}$ symbol processing period, in the j^{th} word, a good filtered estimate of the identification vector, $\underline{B}_j^{i-1}(k)$ is available as $\hat{\underline{B}}_j^{i-1}(k)$. Then, equation (4.4.-9) transforms to

$$S = \prod_{i=1}^{N} \int p(\underline{Z}_{ji}(k)|B_{ji}(k), m_{ji}(k), \underline{Z}_j^{i-1}(k), \hat{B}_j^{i-1}(k), \underline{W}_j^{i-1}(k), \underline{\mathcal{Z}}_{j-1}, \cdot$$

$$\cdot \hat{\underline{w}}_{j-1}, \hat{\underline{B}}_{j-1}) \cdot p(B_{ji}(k)|m_{ji}(k), \underline{Z}_j^{i-1}(k), \hat{\underline{B}}_j^{i-1}, w_j^{i-1}(k), \underline{\mathcal{Z}}_{j-1}, \cdot$$

$$\cdot \hat{\underline{w}}_{j-1}, \hat{\underline{B}}_{j-1}) \, d\, \underline{B}_{ji}(k) \cdot p(m_{ji}(k)|\underline{w}_j^{i-1}(k), \hat{\underline{w}}_{j-1}) \qquad (4.4.-10)$$

Equation (4.4.-10) is a general averaging algorithms which requires a one-symbol conditional-mean prediction density on the identification vector, $\underline{B}_{ji}(k)$. The algorithm is not, however sample recursive, but, rather, processes the received data in symbol blocks.

To obtain a fully recursive form, it is assumed that good estimates are available for filtered and one-sample-stage predicted identification vectors, $\hat{\underline{\beta}}(k)$ and $\hat{\underline{\beta}}(k|k-1)$. Then, (4.4.-10) yields

$$S = \prod_{i=1}^{N} \prod_{k=1}^{N} p(\underline{z}(k)|\hat{\underline{\beta}}(k|k-1), \hat{\underline{B}}_{ji}(k-1), \hat{\underline{B}}_j^{i-1}, m_{ji}(k), \underline{w}_j^{i-1}, \underline{Z}_{ji}(k-1), \cdot$$

$$\cdot \underline{\mathcal{Z}}_{j-1}, \underline{W}_{j-1}, \hat{\underline{B}}_{j-1}) \cdot p(m_{ji}(k)|\underline{w}_j^{i-1}, \hat{\underline{w}}_{j-1} \qquad (4.4.-11)$$

Equation (4.4.-11) is a fully recursive algorithm which represents Q^L parallel branches, one for each word reference $\underline{w}_j^N(k)$. For relatively short code words, (4.4.-11) is a feasible algorithm. For long code words, it is probably more sensible to make a "soft" decision on the symbol sequence, as it progresses. This soft decision is used only for the purpose of reinitializing the filters after each symbol. The resulting algorithm yields M parallel decision-directed branches. However, the word decision statistic still contains Q^L points. The Q^L points are the reduced set obtained from the M^N possible products of the symbol-by-symbol outpout statistic, using only admissible modulator input code words.

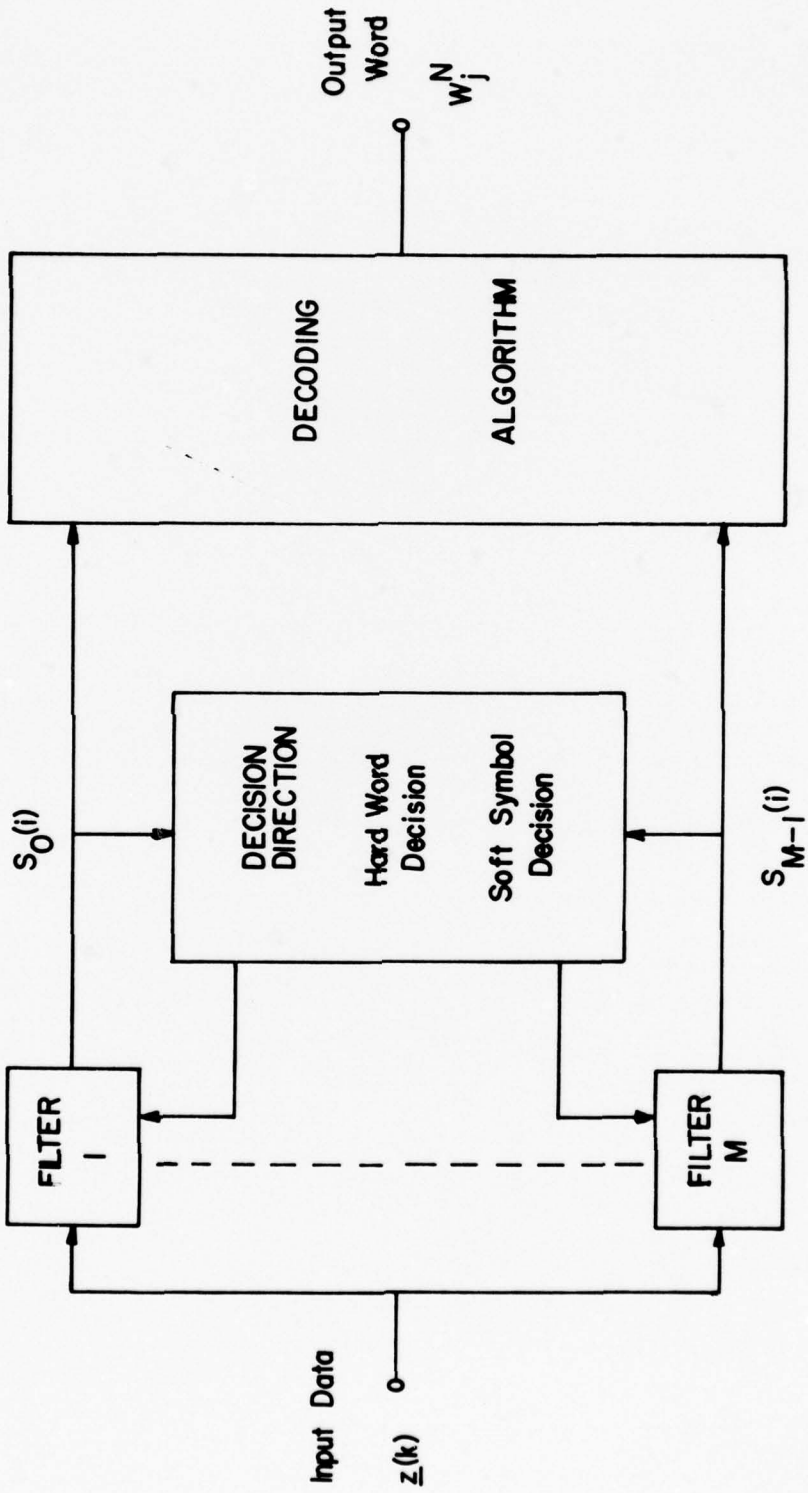


Figure 4.4.-2. General Form of Demodulator/Decoder.

In (4.4.-11), the $p(\underline{z}(k)|(\,)\,)$ are conditionally Gaussian. Thus, the basic structure of the parallel branches consists of paired linear (Kalman) filters and conditional mean identifier predictor-filters, as in Figure 4.1.-3. Following the paired filters is the same multiplicative accumulator for building up the statistic at the end of each symbol period. A new feature, now, in the soft decision case, is a subsequent product accumulator, following the paralleled branches, for building up the statistic at the end of each subsequence of symbols, composing the code word.

4.4.2. Convolutional Coding.

In the previous section, above, has been shown a method for integrating the block coding, decoding function into the detection/estimation/identification problem. The algorithm, so developed, utilizes probabilistic decoding in a natural, yet rigorous, way. It is conjectured with high confidence that convolutional coding, decoding may be integrated in a similar manner. It is further conjectured that the generic minimum form for IDEI Receiver/Decoder will be of the form of Figure 4.4.-2. The generic Receiver/Decoder will process input data, one sample at a time. The parallel branches will deliver decision statistics to a decoder at the end of each symbol. Soft symbol decisions will be used to reinitialize each parallel branch. The decoding algorithm will make a word decision at the end of each N-symbol sequence. The last symbol of the hard word decision will be used to reinitialize the parallel branches and the decoder for processing the next word.

4.4.3. Waveform Design.

It is not enough to just implement coding and probabilistic decoding. Maximization of IDEI receiver performance requires that special attention be paid to the selection of the transmitted waveform, $y_t(t; m_{ji})$, corresponding to m_{ji}, which is the i^{th} symbol in the j^{th} N-sequence. Since bandwidth expanding modulation is not necessarily required, it may be desirable to use narrowband signalling with finite bandwidth waveforms (See Reference [40]). To

circumvent the possibility of intentional interference by waveforms correlated with the transmitted waveform, a selection of waveforms may be necessary, with different waveforms corresponding to different symbols in the D_M alphabet. The trade-off between minimization of correlated interference and the size of the D_M alphabet should also be examined.

Although no effort in the signal design area was made during the period of the Contract reported herein, the area is one which must be investigated if IDEI reception is to be developed for practical use. More is said on this below where initial simulation results are detailed.

SECTION V

THE IDENTIFICATION PROBLEM

5.1. IDENTIFICATION FOR THE DETECTION PROBLEM.

Equation (4.1.-12), above, presented the algorithm for formation of the MAP detection statistic, using Decision-Directed Interference Cancellation and Channel Identification. In derivation of that algorithm, it was seen that an estimation algorithm was required for $\underline{\beta}(k)$, the vector of channel components to be identified. In particular, recursive estimates, $\hat{\underline{\beta}}(k|k-1)$ and $\hat{\underline{\beta}}(k|k)$, were required, which were, respectively, a one-stage conditional-mean predicted estimate, and a subsequent filtered estimate. The fact that conditional-mean estimates were specified by the optimum detection algorithm implies that the optimum identification algorithms are of the type which minimize *the mean of the square of the estimation error.*

The precise forms of the estimators, $\hat{\beta}(k|k-1)$ and $\hat{\beta}(k|k)$, are dictated by the probabilistic models chosen to represent the various components to be identified. With respect to the canonical model of Figure 3.6.-2, the Identification components are partitioned as in Table 5.1.-1.

Now, all of the components listed in Table 5.1.-1, except Δ and $\Delta\omega_j$ are needed in order to optimize (in the Kalman-Wiener sense) the linear filtering algorithms of Section 4.2. The linear filters, in turn, attempt to recreate the Doppler-spreading, Delay-spreading, and additive colored interference. The parameters, Δ and $\Delta\omega_j$, are needed to recreate the deterministic parts of the received signal envelopes. The first question to be asked about the Identification Problem is, "How accurately need the components be identified?" The second question to be asked is, "How is the received I-Q data to be processed to identify the components?"

	Identification Components		Relation to Channel Model
(i)	$\{\Gamma_r, \Phi_r, \Lambda_r\}$ $\{\Gamma_h, \Phi_r, \Lambda_h\}$ $\{\Gamma_j, \Phi_j, \Lambda_j\}$	;	Filter structure for generating Doppler-spreading, Delay-spreading, and additive colored interference.
(ii)	$\underline{\mu}_r, \underline{\mu}_j, \underline{\mu}_t$	;	The strengths of the coherent parts of the direct path, reflected path, and interfering signal.
(iii)	S/N	;	Additive white noise level.
(iv)	Δ	;	Differential delay time between direct path and minimum-delay reflected path.
(v)	$\Delta\omega_j$	;	Frequency offset between direct-path signal and colored interfering signal.

Table 5.1.-1. Identification Components.

5.1.1. Sensitivity of Detection to Identification.

The Kalman and Wiener filters are minimum-mean-squared error filters. That is, when the set $\{\Gamma', \Phi, \Lambda, \underline{\mu}, S/N\}$ are correctly identified and the filter gain function, G, is correctly computed, the Innovations variance is minimized. If $\{\Gamma, \Phi, \Lambda\}$ are set in to the filter slightly in error, the effect is analogous to implementing a filter whose bandwidth is slightly in error. This increases the mean-squared filtering error and Innovations variance above its minimum value. If S/N is incorrectly set in to the filter, the gain function, G, is in error. If $\underline{\mu}$ is incorrectly set into the filter, the "d.c. response" of the filter is in error. When such errors are slight, the effect is to just increase the filter tracking error above its minimum value. When such errors are gross, the filter may diverge or become "un-locked," to borrow a phase-locked loop analogy.

The performance criterion for the linear tracking filters in mean-squared-error. However, the performance criterion for the detection algorithms is symbol error rate, or probability of error. The detection statistic, formed in (4.1.-12), is a functional of the sample variance of the tracking error (Innovations). In the example of Section 4.1.2, the detection statistic was the sample tracking error variance, itself. Thus, identification error, and resulting sub-optimum implementation of the tracking filters, affects the error rate performance measure indirectly. It is not known, based on rigorous analysis, which of the various identification components has the greatest effect on the detection error rate. Consequently, it is not known which of the identifications need to be made most accurately, and to what level of accuracy.

An attempt will be made, in the follow-on to the present contract, to answer some of the questions about the sensitivity of the detection algorithm performance to identification accurancy. The mode of analysis will be mostly through Monte Carlo simulation, although some mathematical analyses will be made, where feasible. Also, various identification algorithms will be investigated for use with the interference tracking detection algorithms.

5.1.2. Implementing Identification with Detection.

It is well known that the conditional-mean predictors and filters, for those components admitting to Gaussian models, are linear, and are, in fact, obtainable from a Kalman filter. However, A Priori, some of the components seem far removed from Gaussian processes. For example, the signal to noise ratio, S/N, for the additive white noise, contains only positive numbers which are as likely to be of the order 10^{-1} as of the order of 10^{-6}. Thus, lacking any prior knowledge of S/N, its natural model seems more likely to fit a uniform distribution than a Gaussian distribution. A similar comment also applies to $\hat{\underline{\mu}}(k)$, the vector of coherent component levels.

The fact that some of the Identification components in $\underline{\beta}(k)$ do not fit Gaussian models, A Priori, implies that some of the associated optimum conditional-mean estimators are non-linear. The general problem of non-linear, conditional-mean recursive estimation is one of current interest. (See Sage and Melsa [41] for a bibliography.) The problem is also one which has yielded few solutions to date.

A two-pronged approach seems indicated for the problem of Recursive Conditional-Mean Identification. The first, or frontal approach, is that of modeling the various components of $\underline{\beta}(k)$ and seeking true conditional-mean recursive estimators for the components thereof. The second, or flanking approach, is described as follows.

The process of identification may be partitioned into two sequential processes, or modes. The first is a Learning, or Acquisition, mode. The second is an Up-dating, or Tracking, mode. During the Learning mode, Maximum Likelihood estimators are used to estimate those Identification components which are, A Priori, non-Gaussian distributed. In parallel with each ML estimator is a second ML estimator for the variance of the ML estimate, itself. For example an ML (vector) estimator would process the data vector, $\underline{z}(k)$, in a manner analogous to that described in [16], to produce an ML estimate, $\hat{\underline{\mu}}(k)$. Also, in parallel, a second ML (matrix) estimator would process the data vector, $\underline{z}(k)$, and the estimate, $\hat{\underline{\mu}}(k)$, to produce the variance matrix for $\hat{\underline{\mu}}(k)$.

As the Learning mode progresses, the ML variance estimates should converge to a minimum. This minimization of the estimate may be anticipated, even if the Identification vector is not stationary. When the absolute rate of decrease of the variance estimate is suitably small, it may be inferred that the non-Gaussian components of $\underline{\beta}(k)$ have been "learned" to the best extent possible, using ML estimators. When the last component has been so learned, the Identification mode passes from Learning to Up-dating.

For the Up-dating mode, those ML estimators for non-Gaussian components of $\underline{\beta}(k)$ are replaced by linear (Kalman) conditional-mean estimators. The linear estimators are based on Gaussian models for the A Priori non-Gaussian distributed components, using the mean and variance obtained from the ML estimators. The ML estimators continue to operate in parallel with the linear estimators, in case the mean and variance change, due to non-stationarity of $\underline{\beta}(k)$. The above is a heuristic, sub-optimum method, reminiscent of the Linearized Kalman filter [41]. However, its performance may be quite satisfactory, in terms of the basic detection problem.

The measure for evaluating the performance of the combined ML-Linear Identifiers, proposed above, is the Error Rate produced by the Integrated Detection/Estimation/Identification algorithms. It was reported in [29] that

improper performance of identification estimators resulted in saturation of the Error Rate at an "irreducible" level. It is conjectured that such irreducibility is partially caused by the identifier algorithm not being properly matched to the underlying stochastic model which produces the component to be identified. Thus, it is conjectured that the proposed combined ML-Linear Identifier, though sub-optimum, should reduce the saturation level of the Error Rate from that previously reported. A sufficient reduction of the saturation level, to, say 10^{-6}, would be practically equivalent to removing the saturation entirely, in view of the possibility of coding implementation described above.

5.2. IDENTIFICATION FOR KALMAN-WIENER FILTERING.

The Identification problem, which has been encountered as a natural adjunct to the interference-tracking detection technique, is a special case of a more general discipline called System Identification [42-45]. The broad definition of Identification is simply the determination of physical models which could produce observed random data. Such physical models may not even be unique. The discipline finds application not just in communications or electrical engineering, but in all branches of modern science and engineering dealing with the analysis of data.

Identification has become important in electrical engineering with the advent of the Kalman filter. Unlike the Wiener filter, synthesis of a Kalman filter requires knowledge of a model for the generation of the observed data in the general canonical form of Figure 5.2.-1. The Wiener filter does not require such specific knowledge, in general. Rather, all that is required is knowledge of the signal mean (d. c. value), $\underline{\mu}(k)$, and auto-covariance function, $V_{yy}(j)$, where

$$V_{yy}(j) = E\{\underline{y}(k + j)\, \underline{y}^T(k)\} \qquad (5.2.-1)$$

Also required is knowledge of the white noise variance, $V_{nn}(0)$.

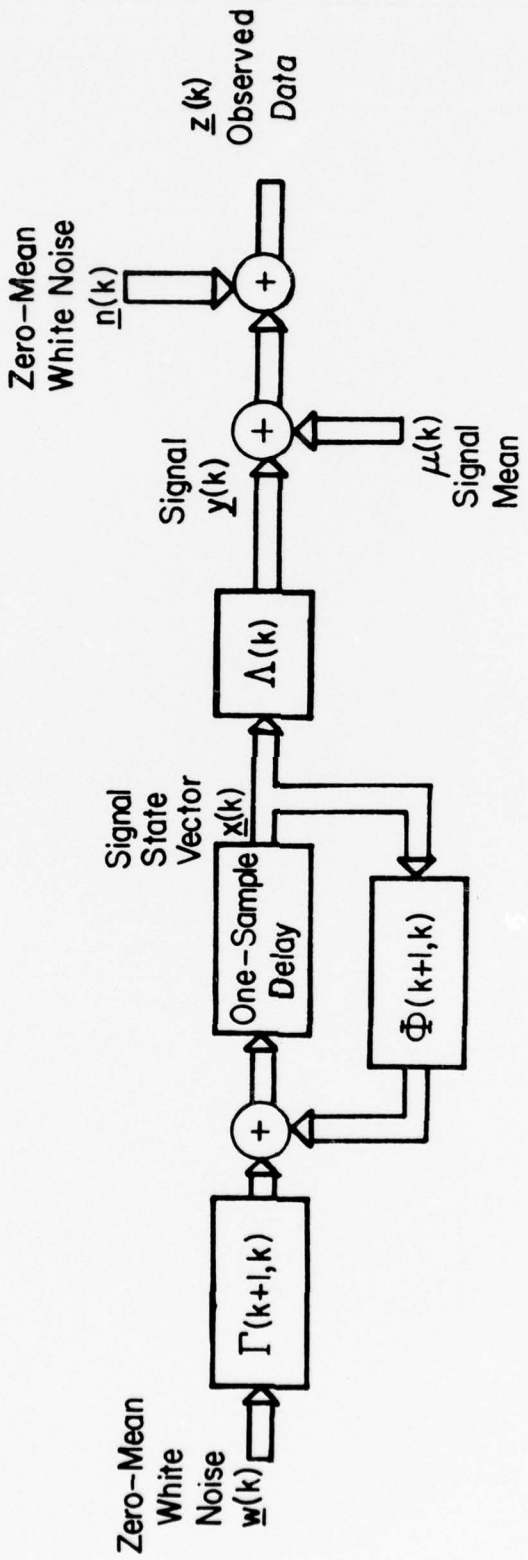


Figure 5.2.-1. Data Generator for Kalman Filter.

The reason that more specific knowledge of a signal generating model is required as in Figure 5.2.-1, is that the Kalman filter is, in general, a time-varying filter for random signals which have time-varying statistics. Thus, the matrices, $\Gamma()$, $\Phi()$, $\Lambda()$, are all generally functions of time, and knowledge of their variation with time is required. If $\Gamma()$, $\Phi()$, and $\Lambda()$ are constant, as in the underlying detection problem above, and if all that was desired was to track the signal $\underline{y}(k)$, or $\underline{y}(k) + \underline{\mu}(k)$, then a standard Wiener filter would suffice. However, the detection statistic requires the Innovations process, $\underline{\nu}(k)$. Thus, the Wiener filter must be in negative feedback canonical form. Hence, the steady-state version of the Kalman filter might as well be used for the Wiener filter.

Fortunately, knowledge of a unique generating model for the data is not required for the Kalman filter. It is easy to show that any set of generator functions $\{\Gamma, \Phi, \Lambda\}$ which are obtained by a similarity transformation (invertible linear transformation) on the state vector, $\underline{x}(k)$, will generate the same output covariance function, $V_{yy}(j)$. From this follows Athans' result [16] that the generator model needed for the Kalman filter is *not* unique. Stronger results by Anderson and Moore [14] and Son and Anderson [15] showed that, only the non-stationary mean, $\underline{\mu}(k)$, and covariance function, $V_{yy}(k+j, k)$, along with the noise variance, $V_{nn}(k)$, are needed to synthesize the non-stationary Kalman filter. Given the covariance function, $V_{yy}(k+j, k)$, the problem of finding a canonical model which can generate it is one of Covariance Factorization [46]. However, in the problem at hand, the covariance function is not given.

The general Identification problem for synthesizing optimum linear filters is, therefore, to determine directly from the observable data, $\underline{z}(k)$, the various structural components required in the Kalman filter. These elements are shown in Figure 5.2.-2, which is the Kalman (or Wiener) filter corresponding to the data generator of Figure 5.2.-1. It is seen from Figure 5.2.-2 that the filter elements which must be identified are $\Lambda(k)$, $\Phi(k+1, k)$, and the filter gain function, $G(k)$. Note that $\Gamma(k+1, k)$ is not necessarily identified directly. Actually, as can be seen from the Kalman gain equations of (4.2.-3), $\Gamma(k+1, k)$ is only used to compute $G(k)$ and enters the computation as $\Gamma\Gamma^T$, rather than directly.

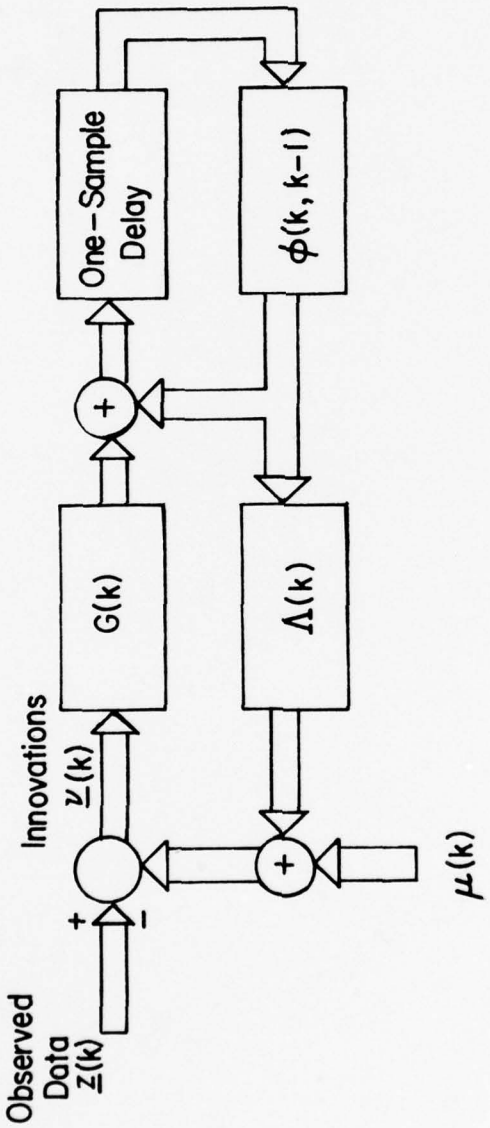


Figure 5.2.-2: Kalman Filter.

The Identification problem is most easily attacked when the signal generator of Figure 5.2.-1 represents a Single-Input-Single-Output (S.I.S.O.) filter. That is, the white scalar input is w(k) and the correlated signal output is y(k). The data is then a scalar, z(k). Much of the earlier theoretical work on Identification was for the S.I.S.O. case.

The problem which arises in identification for IDEI processing is that of identifying a generator structure which is Multi-Input-Multi-Output (M.I.M.O.). In particular, the processes to be identified are the low-pass I-Q processes which are derived from the bandpass additive colored interference or from the low-pass complex Doppler-spreading process. Thus, in Figure 5.2.-1, the white $\underline{w}(k)$ is a 2-vector and the correlated signal output, $\underline{y}(k)$, is a 2-vector, as is the data vector, $\underline{z}(k)$. The M.I.M.O. identification problem is considerably more difficult than the S.I.S.O. problem, as is detailed below.

5.3. IDENTIFICATION OF I-Q PROCESSES.

5.3.1. The I-Q Statistics.

To examine the statistical relations for the In-phase and Quadrature low-pass processes, consider the band-pass formulation, thereof. Define a bandpass Gaussian process, r(t), as

$$r(t) = r_i(t) \cos \omega_c t - r_q(t) \sin \omega_c t \qquad (5.3.-1)$$

which has power spectral density, $S_{rr}(\omega)$, as sketched in Figure 5.3.-1. The power spectral density is the Fourier transform of an autocorrelation function, $R_{rr}(\tau)$, and is therefore real, positive, and an even function of ω. However, note that the density need not be locally symmetric with respect to the frequency, ω_c.

Now, $r_i(t)$ and $r_q(t)$ are the low-pass Gaussian In-phase and Quadrature components of r(t), respectively. Assuming that r(t) is zero-mean, then $r_i(t)$ and $r_q(t)$ are zero-mean and are completely described by their autocorrelation and cross-correlation functions, $R_{ii}(\tau)$, $R_{qq}(\tau)$, and $R_{qi}(\tau)$. For r(t) to be stationary it is necessary and sufficient for [47]

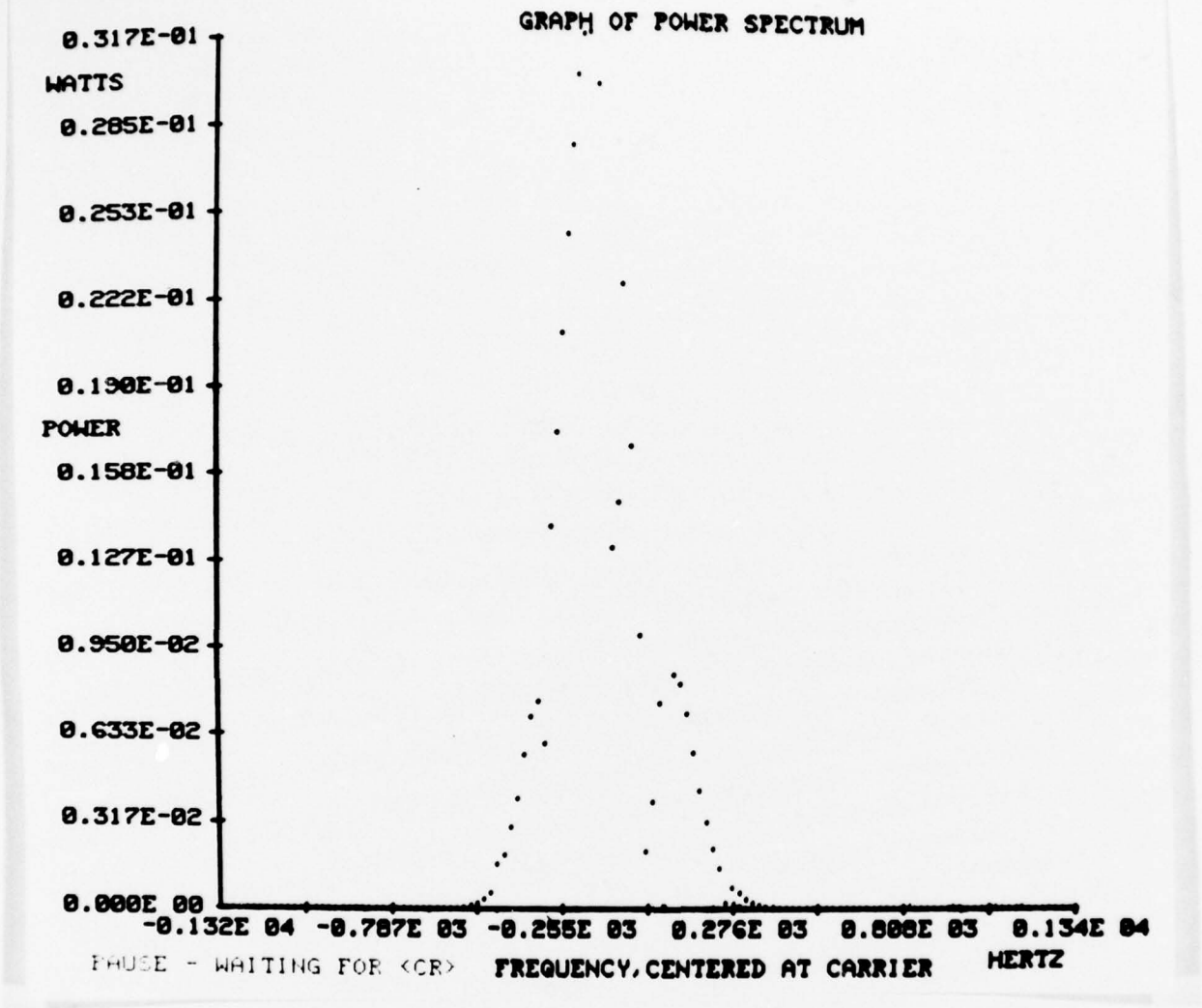


Figure 5.3.-1/ Bandpass Spectral Density

$$R_{ii}(\tau) = R_{ii}(-\tau) = R_{qq}(\tau) = R_{qq}(-\tau)$$

$$R_{qi}(\tau) = R_{iq}(-\tau) = -R_{qi}(-\tau) \qquad (5.3.-2)$$

That is, $R_{ii}(\tau)$ and $R_{qq}(\tau)$ are even functions and $R_{qi}(\tau)$ is an odd function. Note that it is not required for $r_i(t)$ and $r_q(t)$ to be orthogonal (or independent).

The power spectral densities for $r_i(t)$ and $r_q(t)$ are defined by the Fourier transforms

$$S_{ii}(\omega) = F\{R_{ii}(\tau)\}$$

$$S_{qq}(\omega) = F\{R_{qq}(\tau)\}$$

$$S_{qi}(\omega) = F\{R_{qi}(\tau)\} \qquad (5.3.-3)$$

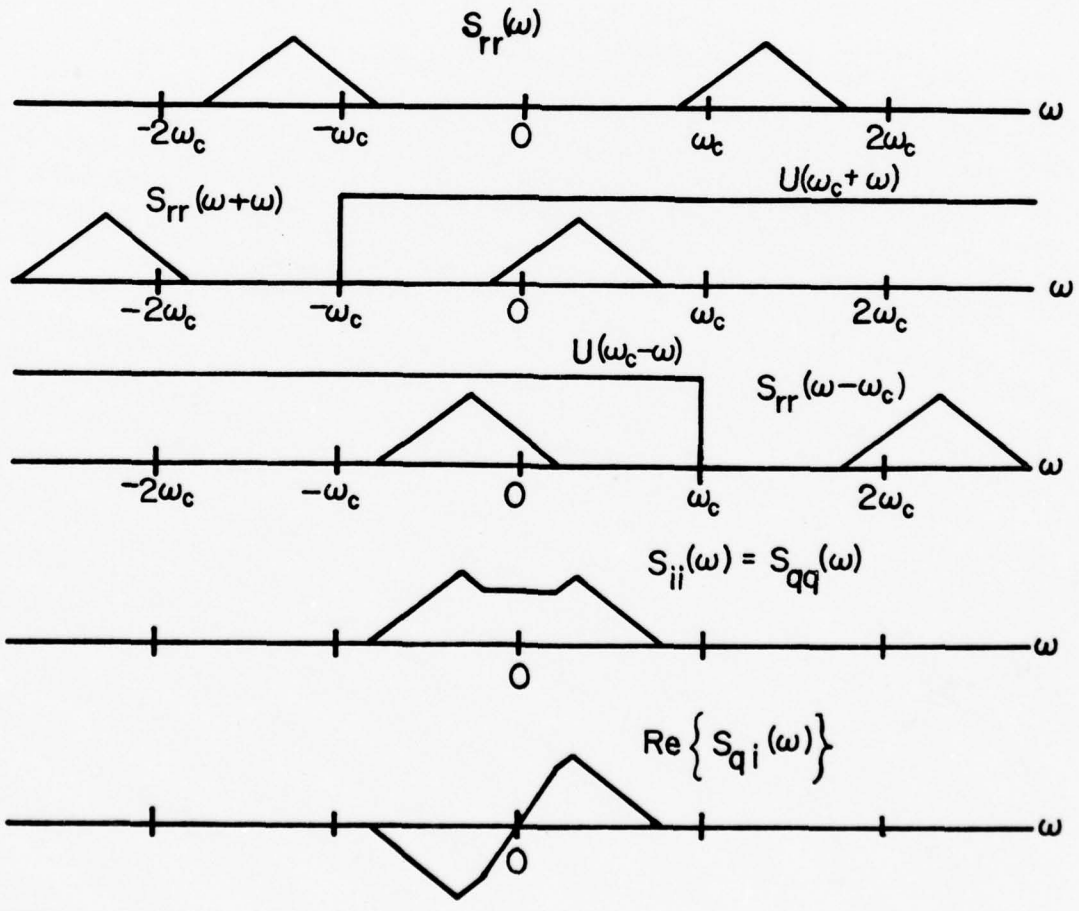


Figure 5.3.-2: Spectral Relations

Since $R_{ii}(\tau)$ and $R_{qq}(\tau)$ are real and even, $S_{ii}(\omega)$ and $S_{qq}(\omega)$ are real, even, and positive. Since $R_{qi}(\tau)$ is real and odd, $S_{qi}(\omega)$ is imaginery and odd.

Now, $S_{ii}(\omega)$, $S_{qq}(\omega)$, and $S_{qi}(\omega)$ may be determined directly from $S_{rr}(\omega)$ by [47]

$$S_{ii}(\omega) = S_{rr}(\omega_c + \omega)\, U\,(\omega_c + \omega) + S_{rr}(\omega - \omega_c)\, U\,(\omega_c - \omega)$$

$$S_{qi}(\omega) = j[S_{rr}(\omega - \omega_c)\, U\,(\omega_c - \omega) - S_{rr}(\omega_c + \omega)\, U\,(\omega_c + \omega)] \qquad (5.3.-4)$$

where U() is the Unit-Step function defined by

$$U(x) = 1 \quad ; \quad 0 \leq x$$

$$= 0 \quad ; \quad x < 0 \qquad (5.3.-5)$$

Figure 5.3.-2 shows the various spectral relations from (5.3.-4) for a hypothetical non-symmetric bandpass power spectral density. Note that if $S_{rr}(\omega)$ were locally even symmetric with respect to the frequency, ω_c, then $S_{qi}(\omega)$ would be identically zero. Hence, $R_{qi}(\tau)$ would be identically zero (limiting case of an odd function). Thus, $r_i(t)$ and $r_q(t)$ would be orthogonal, uncorrelated, and independent (since they are Gaussian).

If $r_i(t)$ and $r_q(t)$ were independent, they could then be identified independently, using S.I.S.O. identification techniques. Unfortunately, in general, the Doppler spectra or additive colored interference spectra are not locally even symmetric. Thus, in general, $r_i(t)$ and $r_q(t)$ are correlated, and M.I.M.O. identification must be used.

5.3.2. The I-Q Generator Model.

For the present problem, the I-Q Generator Model is the stationary version of that of Figure 5.2.-1, for which $\{\Gamma(k+1, k), \Phi(k+1,\Lambda(k), \underline{\mu}(k)\}$ are either constant or so slowly time-varying that they may be taken as constant for the purposes of recursive identification. This means that the elements would be

constant over the interval of time corresponding to the memory of the identifying algorithms.

Thus, the I-Q model is defined by the equations

$$\underline{X}(k+1) = \Phi\underline{X}(k) + \Gamma\underline{w}(k)$$

$$\underline{y}(k) = \Lambda\underline{X}(k) + \underline{\mu}(k)$$

$$\underline{z}(k) = \underline{y}(k) + \underline{n}(k) \qquad (5.3.-6)$$

Now, $\underline{y}(k)$, $\underline{n}(k)$, $\underline{z}(k)$, and $\underline{w}(k)$ are all 2-vectors. In terms of the I-Q formulation, they are defined by

$$\underline{y}(k) = \begin{bmatrix} y_i(k) \\ \\ y_q(k) \end{bmatrix} \quad ; \quad \underline{n}(k) = \begin{bmatrix} n_i(k) \\ \\ n_q(k) \end{bmatrix} \quad ; \quad \underline{\mu}(k) = \begin{bmatrix} \mu_i(k) \\ \\ \mu_q(k) \end{bmatrix}$$

$$\underline{z}(k) = \begin{bmatrix} z_i(k) \\ \\ z_q(k) \end{bmatrix} \quad ; \quad w(k) = \begin{bmatrix} w_i(k) \\ \\ w_q(k) \end{bmatrix} \qquad (5.3.-7)$$

Both $\underline{n}(k)$ and $\underline{w}(k)$ are independent, zero-mean, white and Gaussian. It is assumed that $y_i(k)$ and $y_q(k)$ are each Markov-N. Thus, the state-vector, $\underline{X}(k)$, must have 2N states, and the obvious definitions follow:

$$\underline{X}(k) = \begin{bmatrix} \underline{X}_i(k) \\ \hline \underline{X}_q(k) \end{bmatrix} \quad ; \quad \Phi = \left[\begin{array}{c|c} \Phi_{ii} & \Phi_{ig} \\ \hline \Phi_{qi} & \Phi_{qq} \end{array}\right]$$

$$\Gamma = \left[\begin{array}{c|c} \gamma_{ii} & \gamma_{ig} \\ \hline \gamma_{qi} & \gamma_{qq} \end{array}\right] ; \quad \Lambda = \left[\begin{array}{c|c} \underline{\lambda}_{ii}^T & \underline{\lambda}_{ig}^T \\ \hline \underline{\lambda}_{qi}^T & \underline{\lambda}_{qq}^T \end{array}\right] \qquad (5.3.-8)$$

where the $\underline{X}_i(k)$, $\underline{X}_q(k)$, $\underline{\gamma}_{ii}$, $\underline{\gamma}_{iq}$, $\underline{\gamma}_{qi}$, $\underline{\gamma}_{qq}$, $\underline{\lambda}_{ii}$, $\underline{\lambda}_{iq}$, $\underline{\lambda}_{qi}$, and $\underline{\lambda}_{qq}$ are all N-vectors. The Φ_{ii}, Φ_{iq}, Φ_{qi}, and Φ_{qq} are each N x N matrices.

Identification of the I-Q model involves three interacting steps: (i) Selection of a structure for $\{\Gamma, \Phi, \Lambda\}$ which contains a minimum number of non-zero elements; (ii) Identification of the order, N, of the model; (iii) Identification of the individual non-zero elements in the set $\{\Gamma, \Phi, \Lambda\}$. Note that in the general model of (5.3.-8) the number of elements to be identified is $(2N)^2 + 4(2N)$. The question is, "How many of the elements can be set to zero, A Priori, in selecting a minimum structure for $\{\Gamma, \Phi, \Lambda\}$?"

M.I.M.O. Identification has been investigated in general, but not for the specific model which results for the I-Q case. Popov [48] has developed a complete set of invariants for a controllable pair. Weinert and Anton [49], Denham [50], and Sinha and Rozsa [51] have extended Popov's work to develop invariant canonical forms suitable for M.I.M.O. Identification. Mehra [52] and Tse and Weinert [53] have shown methods of identifying the order, N, using the Innovations process (see Kailath and Frost [54]).

Results for M.I.M.O. Identification algorithms are few. Tse and Weinert [53] have used given output covariance functions to identify Φ, Λ, and V_{nn}, but not Γ, using the Luenberger canonical form. Their resulting algorithm was not recursive. Graupe, et al., [55] have identified the order, N, using Autoregressive Moving Average (ARMA) formulations. The resulting algorithm has very complicated structure. Kashyap [56], has used ARMA models for Identification, as have Rowe [57] and Mayne [58], who have also transformed the results into a state-variable canonical form. Martin and Stubberud [59] have also identified parameters and covariances using the Innovations process.

The chief difficulties with previous M.I.M.O. Identification results are that (i) they do not apply directly to the I-Q case; (ii) they are not recursive; and (iii) they are overly complex. What is first necessary is to determine the minimum model for the I-Q case, given order, N. Having the minimum model, then, recursive Identification algorithms must be developed.

The covariance relations for the general model are

$$E\{[\underline{z}(k+j) - \overline{\underline{z}}(k+j)][\underline{z}(k) - \overline{\underline{z}}(k)]^T\} \triangleq V_{zz}(k+j, k) =$$

$$= V_{yy}(k+j, k) + V_{nn}(k+j, k)$$

$$V_{yy}(k+j, k) = \Lambda V_{XX}(k+j, k)\Lambda^T$$

continued

$$V_{XX}(k+j,\ k) = \Phi^{j} V_{XX}(k,\ k)$$

$$V_{XX}(k+1,\ k+1) = \Phi V_{XX}(k,\ k)\Phi^{T} + \Gamma\Gamma^{T}$$

$$V_{nn}(k+j,\ k) = V_{nn} \cdot \delta_{oj} \qquad (5.3.-9)$$

where $\underline{\bar{z}}(k)$ is the mean of $\underline{z}(k)$ given as

$$\underline{\bar{z}}(k) = E\{\underline{z}(k)\} = \underline{\mu}(k) = \underline{\mu} = \begin{bmatrix} \mu_i \\ \mu_q \end{bmatrix} \qquad (5.3.-10)$$

In (5.3.-9), $V_{nn}(k+j,\ k)$ is the covariance of white stationary noise, which is the constant V_{nn} matrix times the Kroneker delta, δ_{oj}. It is assumed that $\underline{w}(k)$ has unit variance. It is assumed that the generator has reached steady state and that $\underline{X}(k)$ is stationary. Under these conditions, equations (5.3.-9) become

$$V_{zz}(k+j,\ k) = V_{zz}(j) = V_{yy}(j) + V_{nn}\delta_{oj}$$

$$V_{yy}(j) = \Lambda\Phi^{j} V_{XX}\Lambda^{T}$$

$$V_{XX} = \Phi V_{XX}\Phi^{T} + \Gamma\Gamma^{T} \qquad (5.3.-11)$$

Now, the requirements of equation (5.3.-2) are that for $y_i(k)$ and $y_q(k)$ to be low-pass I-Q components, it must be satisfied that

$$E\{[\underline{y}(k+j) - \underline{\mu}][\underline{y}(k) - \underline{\mu}]^{T}\} \triangleq V_{yy}(j) =$$

$$= E\left\{\begin{bmatrix} y_i(k+j)y_i(k) & y_i(k+j)y_q(k) \\ y_q(k+j)y_i(k) & y_q(k+j)y_q(k) \end{bmatrix}\right\} =$$

$$= \begin{bmatrix} V_{ii}(j) & V_{iq}(j) \\ V_{qi}(j) & V_{qq}(j) \end{bmatrix}$$

continued

$$V_{ii}(j) = V_{qq}(j) \quad ; \quad \text{an even function}$$

$$V_{iq}(j) = -V_{iq}(-j) = -V_{qi}(j) \quad ; \quad \text{an odd function} \qquad (5.3.-12)$$

The structure of $\{\Gamma, \Phi, \Lambda\}$, as given in (5.3.-8) must be chosen to satisfy (5.3.-12) with a minimum number of non-zero elements. The search for such a structure, along with recursive algorithms for identifying it is a part of the follow-on extension to the present contract.

5.4. IDENTIFICATION OF THE MEAN.

5.4.1. Modeling the Mean as Gaussian.

When the mean, $\underline{\mu}$, of equations (5.3.-6) and (5.3.-10) is non-zero, it must be supplied to the optimum linear filter (Kalman or Wiener). The reason for this is that the mean plays the part of a "constant d.c. level" in waveform filtering. Although the optimum linear filter has "d.c. gain," it is not generally unit gain. Therefore, the mean value, $\underline{\mu}$, is not recovered exactly by the filter. More importantly for the detection problem, an unknown mean results in a non-zero mean, or bias, in the Innovations process. Since the Innovations process is used to form the detection statistic, an unknown mean, $\underline{\mu}$, affects the detection error rate immediately by biasing the Innovations.

A common way out of this dilemma is to treat the mean $\underline{\mu}$ as a slowly time-varying Gaussian vector and to augment the state vector, $\underline{X}(k)$, with another state vector from which $\underline{\mu}(k)$ is assumed to be generated. The optimum linear filter then attempts to track both $\underline{y}(k)$ and $\underline{\mu}(k)$. This procedure results in a modeling error in the case where $\underline{\mu}$ is constant, and is therefore sub-optimal. Also, the dimension of the filter's state vector is increased, which is also not desirable.

A better result by Friedland [60] is based on the augmented state-vector approach where $\underline{\mu}$ is modelled as constant Gaussian. This leads to two estimators. The first is the standard linear filter for a zero-mean signal. The second is a linear filter, driven by the biased Innovations process, which produces an optimal estimate of the mean $\underline{\mu}$. Using Friedland's scheme the unbiased true Innovations process may then be recovered for use by the detector.

This technique will be pursued in the follow-on extension to the present contract.

5.4.2. Maximum-Likelihood Mean Identification.

A Maximum-Likelihood mean estimator was reported in [29], based on sample-mean averaging of the Innovations process. This was an Ad Hoc estimator, not based on any rigorous analysis. The performance of the estimator itself was not measured. Rather, the estimator performance was inferred from error rate performance of the detection algorithms simulation with the mean estimator running. In retrospect, it not obvious whether the ML mean estimation algorithms performed well or whether one state of the Kalman filter itself was tracking the mean.

It may be that a good practical approach to the Mean Identification problem will be to augment the state vector with two additional states, one for each component of the mean. The closed loop filter gain for these two states may be set close to unity. This will give an ML-like estimate of the mean. The performance of this kind of Mean Identifier will be examined in the follow-on extension to the present contract.

SECTION VI

THE MONTE CARLO SIMULATION

6.1. PROGRAM DESCRIPTION

6.1.1. Overview

The overall simulation program operates as shown in Figure 6.1.-1. A random Message Generator produces pseudo-random symbols in the {0, 1} alphabet. The occurrence of successive symbols is independent in the statistical sense. The symbols are routed to a Modulation Generator where the appropriate I-Q modulation waveforms are generated as per equations (3.2.-3), (3.2.-4), (3.2.-5) and (3.2.-9).

An E. M. Reflection Routine, developed by Peake [11], produces the delay-spread function, $P(\tau)$, used in (3.3.-17), and the Doppler-sreading spectrum, $S\rho(\omega)$, used in (3.3.-21). The outputs of the E. M. Reflection Routine and Modulation Generator are used in the Channel Processor to produce the components of signal received via the direct and reflected paths. At the present state of development of the Monte Carlo simulation program, the use of the E. M. Simulation Routine outputs is not automated. That is, the components for the structure of the H-filter and R-filter, in Figure 3.6.-2, are not calculated automatically from the outputs of the E. M. Reflection Routine. Rather, the filter components are set into the Channel Processor manually. Such automation is actually a part of the Identification Problem, as discussed in Section V.

A Jamming Generator and White Noise Generator complete the formation of the received data process. The Jamming Generator produces the colored interference signal, as per equations (3.5.-17) through (3.5.-22). The White Noise Generator produces the receiver-generated white noise, as per equations (3.5.-11) through (3.5.-15).

The received data is processed by the Optimum Receiver Algorithms and by the Standard Receiver Algorithms. These algorithms are detailed in Sections 4.1, 4.2, and 4.3. The detected message symbols from both the optimum receiver and standard receiver are then routed to the Error Rate Generator and

Statistical Significance Estimator. There, the detected symbols are compared to the transmitted symbols and errors are recorded. From the errors, error rates are computed, as are the statistical means and variances of the error rates, also.

The above overview of the Monte Carlo simulation program is strictly functional. A detailed description of the actual program and all its elements follows. Program listings are in Section 6.2.

6.1.2. Main Program Description

The flow diagram for the main program is given as Figure 6.1.-2. It is seen that the main program is built around a single DO-LOOP which operates on data sample number. Thus, the total Monte Carlo simulation program operates recursively, sample by sample, just as do the optimum and standard receivers. In this way, statistically valid runs at lower error rates are achieved by running the simulation for a longer time, rather than by increasing the required machine storage. This simulation strategy was adopted originally at NASA Langley Research Center, where long run times on the 6600 were easy to obtain, but large storage was not. Also, this is the obvious strategy for simulation on storage-limited minicomputers.

The following description of the operation of the Main Program is keyed to the flow diagram in Figure 6.1.-2. See also the listing for the Main Program, which is given below in Section 6.2.

The first operations are dimensioning variables and setting up a common block of variables. These variables are described below. Also, there is associated BLOCK DATA, which is detailed below. Next, some housekeeping functions are done for the PDP 11/40, assigning system units to the Teletype keyboard, Card Reader and CRT Control Console. Also, tabulation of elapsed real time as the program executes is facilitated.

The first subroutine to be called is INPUT. This subroutine accepts all the input data to the program such as symbol period, modulation type, signal to noise ratios, etc., and computes the structural elements of the H-filter, R-filter, J-filter, etc. Also, INPUT formats the data print-out page which describes the case being run. Next, formatting and writing is done for the final statistics at the end of the simulation run. Also, the start time is recorded and written.

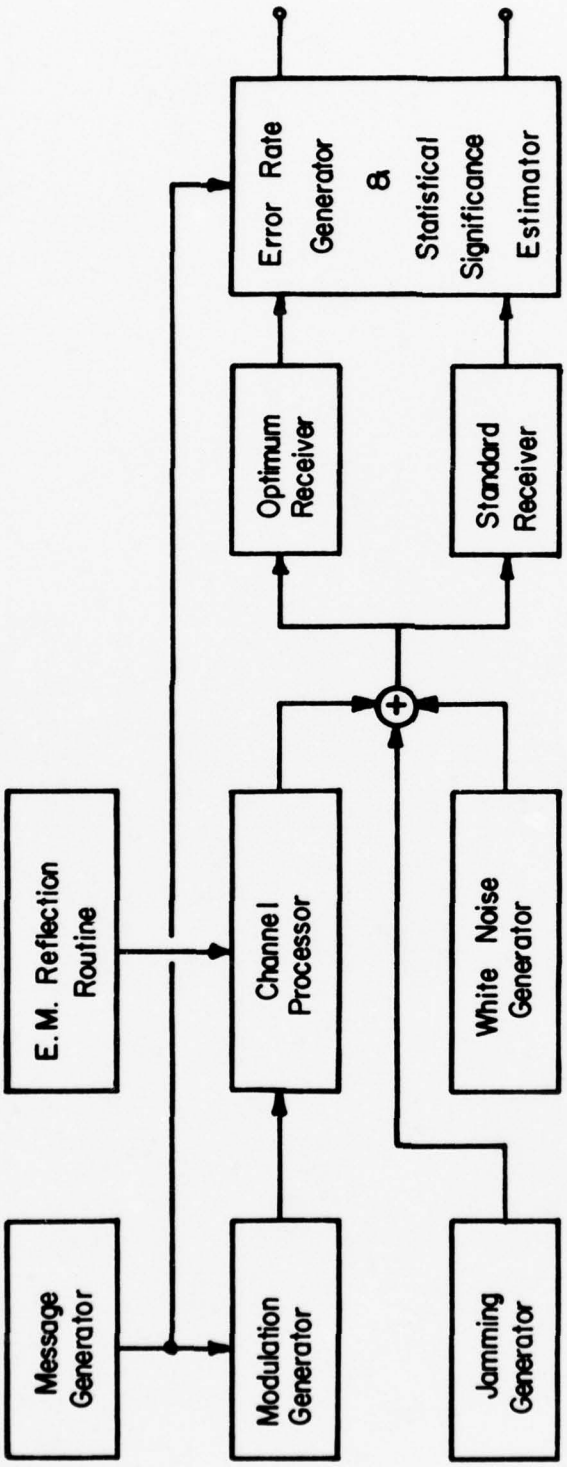


Figure 6.1.-1: Computer Simulation Structure.

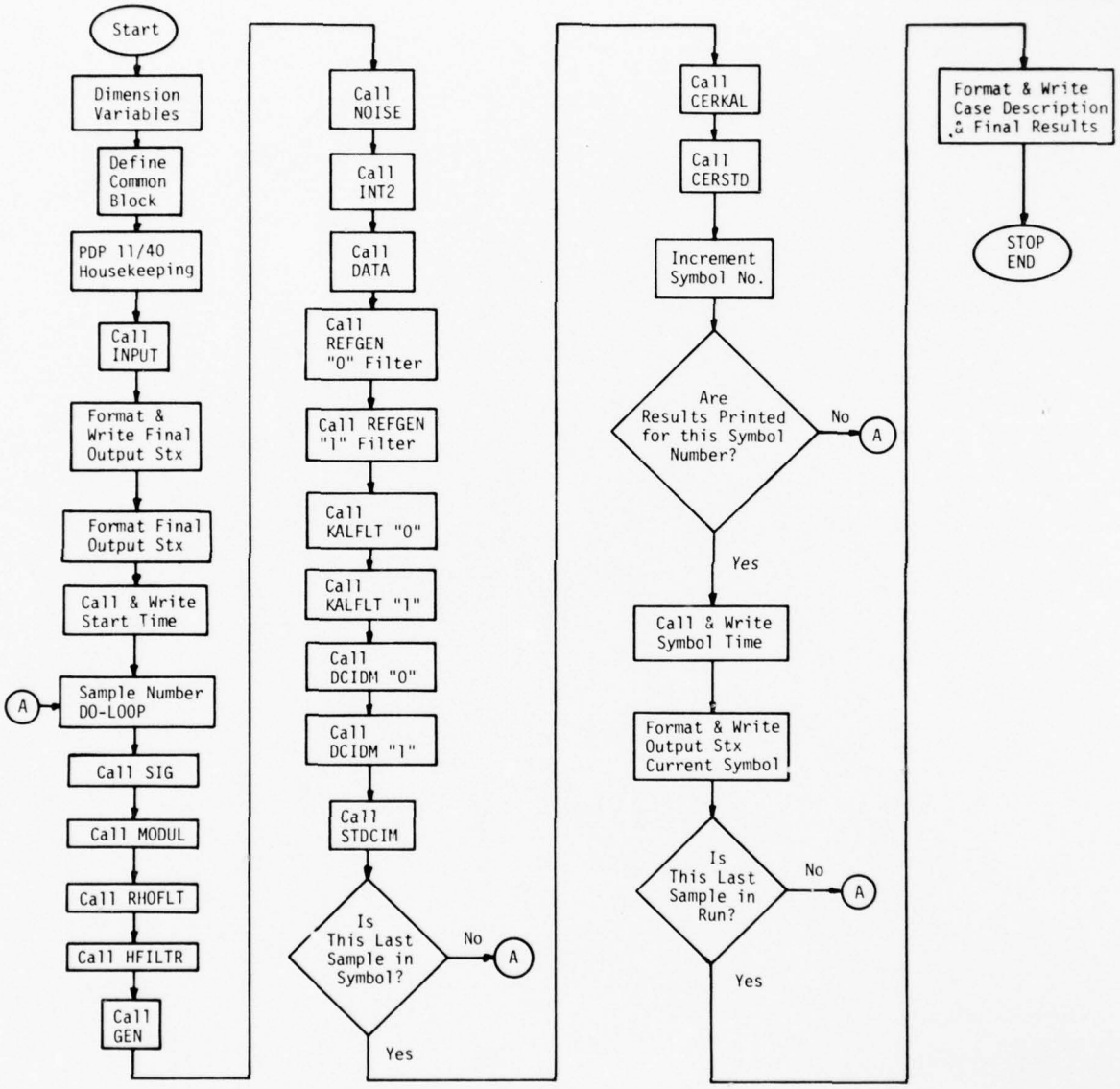


Figure 6.1.-2: Main Program Flowchart.

The run-time of the simulation is controlled by the DO-LOOP on sample number, KS. The generation and processing of data samples is terminated at sample number, NDS, which is an input number. The main body of operations in the DO-LOOP consists of calls to data generating and processing subroutines. These subroutines are takne in order below.

SIG generates the sample value of the direct path symbols, BB, at sample, number KS, in the binary alphabet {0, 1}. SIG also generates the sample value of the differentially delayed symbol, BBR, on the reflected path.

MODUL converts the symbol sample to the alphabet {+1, -1} and then forms the samples of the modulation waveforms in I-Q form, for both the direct path and the differentially delayed reflected path. XST is the I-Q vector corresponding to $\underline{y}'_t(k, m)$ of equation (3.5.-9). FR and GR are the scalars corresponding to $f_t(k; m; \Delta)$ and $g_t(k; m; \Delta)$ in equation (3.5.-10).

RHOFLT creates the multiplicative noise, as in (3.5.-10). The output of RHOFLT is RHO, which is the I-Q two-vector corresponding to $\underline{y}_r(k)$ in (3.5.-10). The variable XKRHO is the state-vector (array) corresponding to $\underline{X}_r(k)$ in (3.5.-10). NRHO is the dimension of either the I state-vector or the Q state-vector (they are the same dimension). NRHO2 is the dimension of the total state vector for the I-Q generator and is equal to twice NRHO.

In the present version of the main program, the next call is to HFILTR which is the H-filter of Figure 3.6.-2. This version of the program is obsolete and has not yet been up-dated to represent the filtering used in (3.5.-10). The model described by the present program is one wherein the differentially-delayed modulation was first filtered in the H-filter and then multiplied by the multiplicative noise. The model which has finally evolved into (3.5.-10) does the multiplication first and the H-filtering second. The effect of the required modifications to the program will be to reverse the order of the calls to HFILTR and to GEN, which is next in the present version.

As long as no delay-spreading is used, the present program and the modified program will operate identically.

Subroutine GEN performs the matrix multiply to give to give the term, $H_t(k; m; \Delta)\underline{\rho}(k)$, in (3.5.-10). The output, XSR, is the I-Q vector corresponding to the above product.

Subroutine NOISE generates the I-Q samples of zero-mean white additive noise, $\underline{n}(k)$, as per (3.5.-11). The output I-Q vector is XN.

Subroutine INT2 generates the colored stochastic additive interference, $\underline{y}'_j(k)$, according to (3.5.-17), (3.5.-18) and (3.5.-22). The I-Q vector output, XJ, corresponds to $\underline{y}'_j(k)$ in (3.5.-17). The state-vector for the generator is XKJ of dimension NJ2, where NJ2 is twice NJ, which is the dimension of the I and of the Q state-vector.

Subroutine DATA forms the I-Q data vector, $\underline{Z}$, from the outputs XST, XSR, XJ, and XN. At this point the received data has been formed. What follows next are the receiver processing operations and the error-rate calculations.

The subroutine REFGEN is called twice. The first time, it generates $H_t(k; m=0)$ and $H_t(k; m=0, \Delta)$. The second time, it generates $H_t(k; m=1)$ and $H_t(k; m=1, \Delta)$. These matrices are the linear filter references required in (4.2.-2). The outputs of REFGEN are FTR, GTR, FRR, and GRR. These are the counterparts of XST, FR, and GR, which were generated by MODUL.

Next, the Kalman filtering routine, KALFLT, is called twice, once for m=0 and once for m=1. The outputs are XEST, VEST, AINOV, VINV, and DET. XEST and VEST are the filtered state estimate and variance respectively which are required in the filter and gain equations as per (4.2.-2) and (4.2.-3) respectively. AINOV is the Innovations process, VINV is the inverse of the Innovations variance and DET is the determinant of the Innovations variance, as required for the detection algorithms of (4.1.-13). Note that in KALFLT, the dimensions of the multiplicative and additive colored noise state-vectors are set as NYRR and NYJR, respectively. These dimensions may be different (due to identification) than the corresponding dimensions, NRHO2 and NJ2, in the respective noise generators.

Following the linear filter, the symbol decision statistics are formed recursively for m=0 and m=1 by two calls to DCIDM. The outputs of DCIDM are SUM and S. For PSK and FSK modulation, the product on k in (4.1.-12) may be converted to a sum, since the Gaussian density function is exponential in form. The running sum is output at each sample time as SUM. At the last sample time in each symbol period, the cumulative sum is output as the final devision statistic, S.

The next step in the program is to call the decision sub-routine for the standard receiver, STDCIM. The standard PSK detector is extremely simple. The detection statistic, SPSK, is simply the cumulative sum of the Q-channel samples. For FSK, the standard detector is somewhat more complicated. The running statistics outputs at each sample time are AFSK and BFSK, which are

the counterparts of a(k; m) and b(k; m) in (4.3.-41). At the last sample time in each symbol period, the final cumulative detection statistic, SFSK, is outputted, corresponding to $a^2(k; m) + b^2(k; m)$ in (4.3.-30). It is seen that the PSK detection sub-routine also uses the waveform references, FTR and GTR.

Next, the sample number, KS, is tested to see if it represents the last sample in a symbol period. If it is not, the main program returns and re-enters the DO-LOOP on KS. If KS is the final sample number for a symbol, the main program calls CERKAL.

Sub-routine CERKAL takes in the detection statistics, S, from both calls to DCIDM, and decides which symbol has been received. From this and the actual transmitted symbol, BB, CERKAL forms a cumulative count, ERR, of detection errors for the optimum detector. CERKAL also provides as output, the raw error rate, ERRATE, an exponentially smoothed mean error rate, ERMEAN, and an exponentially smoothed error rate variance, ERRVAR. Also produced is a measure of confidence in the smoothed error rate, DATSNR, which is the square of ERMEAN divided by ERRVAR. Finally, having decided on the correct symbol, CERKAL resets the final state variable filtered estimate, XEST, and tracking error variance, VEST, in the "incorrect" tracking filter.

Next, subroutine CERSTD is called to make the symbol decision for the standard detector. It also calculates a cumulative error, ERRS, error rate, ERATES, smoothed mean error rate and variance, ERMENS, ERVARS, and data condition, DATSNR. Now, the symbol number in the transmitted sequence is incremented and tested to determine if results are to be printed for this symbol. If not, the program returns and re-enters the DO-LOOP on KS. If results are to be printed (every NPRNT symbols), then GTIM is called to print out clock time. Also printed out are symbol number, IB, and the various error rate statistics.

The main program may be terminated in several different ways, one of which is when the DO-LOOP on sample number, KS reaches sample number, NDS. The other methods of termination are explained below in description of second-level sub-routine, ESTERR.

Following termination, the final data output is written, according to FORMAT statement 200.

Tables 6.1.-1 and 6.1.-2, below describe the DIMENSIONed and COMMON variables, respectively for the Main Program. Additionally, STRNG is an 8-bit (1 byte) logical variable. JTIME is Integer *4. Both are required for calling clock time during print-out.

NAME	TYPE	USES
XKH	Real *4, 1-dimensional array, length 10	H-filter state-vector
XKRHO	Real *4, 1-dimensional array, length 10	R-filter state-vector
XKJ	Real *4, 1-dimensional array, length 10	J-filter state-vector
XEST	Real *4, 1-dimensional array, length 12	Kalman filter state-vector
VEST	Real *4, 2-dimensional array, 12 X 12	Kalman filter variance matrix
FHGH	Real *4, 1-dimensional array, length 2	H-filter output vector
RHO	Real *4, 1-dimensional array, length 2	R-filter output vector
XST	Real *4, 1-dimensional array, length 2	Direct-path vector
XSR	Real *4, 1-dimensional array, length 2	Reflected-path vector
XN	Real *4, 1-dimensional array, length 2	White Noise vector
XJ	Real *4, 1-dimensional array, length 2	Jamming Signal vector
Z	Real *4, 1-dimensional array, length 2	Data vector
AINOV	Real *4, 1-dimensional array, length 2	Innovations vector
VINOV	Real *4, 2-dimensional array, 2 X 2	Innovations Variance matrix
VINV	Real *4, 2-dimensional array, 2 X 2	Inverse of VINOV

Table 6.1.-1. Main Program Dimensioned Variables.

NAME	TYPE	USE
CRTN:		
NDS	Integer *4	Maximum number of data samples
NPRNT	Integer *4	Print results every NPRNT symbols
NIB	Integer *4	Maximum number of message symbols
NCASEQ	Integer *4	Error rate convergence limit
SAMPLE:		
NSPB	Integer *4	Number of samples per bit
TB	Real *4	Symbol period, secs. - Direct path
TBR	Real *4	Symbol period, secs. - Reflected path
RATIO:		
RHODB	Real *4	Reflected to Direct Path Signal Ratio, dB
SNRDB	Real *4	Direct Path Signal to White Noise Ratio, dB
SJRDB	Real *4	Direct Path Signal to Jamming Ratio, dB
RHODBR	Real *4	Kalman Reference for RHODB
SNRR	Real *4	Kalman Reference for SNRDB
SJRDBR	Real *4	Kalman Reference for SJRDB

Table 6.1.-2. Main Program Common Variables.

6.1.3. Block Data (BLKDAT)

This subroutine-like program component stores and/or initializes all common block variables. The SEED integers are required for the random number generating subroutine, RANC, which drives the white Gaussian generator, MARSA. Various initializing seed numbers are stored which have been found, experimentally, to give quick statistical convergence of RANC.

The variables stored under ZPWUP are the $\{\Gamma, \Phi, \Lambda\}$ parameters for the H-filter, R-filter, and J-filter of Figure 3.6.-2. The variables stored under ZPWKAL are the identified versions of the $\{\Gamma, \Phi, \Lambda\}$ parameters which are used to structure the Kalman filter of Figure 4.2.-1. Also stored under ZPWKAL are the identified strengths (amplitudes) of the specular (coherent) components of the reflected signal and jamming signal. Stored under SIGCON are the delayed binary signal samples when differential delay exists between direct and reflected paths, as per equation 3.3.-9.

NAME	TYPE	USES
SEED:		
IXS, JSX	Integer *4	Binary Signal Seeds
IXG1, JXG1, IXG2, JXG2	Integer *4	Mult. Noise Seeds
IXN1, JXN1, INX2, JXN2	Integer *4	White Noise Seeds
IXJ1, JXJ1, IXJ2, JXJ2	Integer *4	Jammer Noise Seeds
ZPWUP:		
GAMH, PHEEH, CH	Real *4, 1-dimensional array, length 5	H-filter parameters
GAMRHD, PHERHO, CAHO	Real *4, 1-dimensional array, length 5	R-filter parameters
GAMJ, PHEEH, CJ	Real *4, 1-dimensional array, length 5	J-filter parameters
ZPWKAL:		
GAMRR, PHEERR, CRR	Real *4, 2-dimensional arrays,	Identified R-filter
GAMJR, PHEEJR, CJR	Real *4, 6x2, 6x6, 2x6, respectively	Identified J-filter
AMUR	Real *4	Identified Reflc. Magnitude
AMUJ	Real *4	Identified Jammer Magnitude
CRTN: Same as Main Program		
SAMPLE: Same as Main Program		

Table 6.1.-3. Block Data Common Variables

NAME	TYPE	USES
DELAY:		
DEL	Real *4	Differential delay, Δ, seconds
DELPHI	Real *4	Phase deviation, $\Delta\phi$, radians
DELMEG	Real *4	Freq. deviation, $\Delta\omega$, radians/second
DELMEJ	Real *4	Freq. offset, $\Delta\omega_j$, radians/second
DELJR	Real *4	Identified Freq. Offset, radians/second
AMAG:		
AT	Real *4	Direct path signal magnitude
AR	Real *4	Reflected path signal magnitude
AEST	Real *4	Identified Direct path magnitude
OPTION:		
NOH	Integer *4	0, No delay spread H-filter 1, Delay spreading with H-filter
NOS	Integer *4	1, Binary PSK 2, Binary FSK
RATIO:	Same as Main Program	

Table 6.1.-3. Block Data Common Variables (Continued)

NAME	TYPE	USES
GDB:		
GRHO	Real *4	Mult. Noise Ampl. Constant
GN	Real *4	White Noise Ampl. Constant
GJ	Real *4	Jam. Noise Ampl. Constant
GRHOR	Real *4	Ident. Mult. Noise Ampl.
GNR	Real *4	Ident. White Noise Ampl.
GJR	Real *4	Ident. Jam. Noise Ampl.
SIGCON:		
SIGB	Real *4, 1-dimensional array, length 50	Delayed Binary Sig.

Table 6.1.-3. Block Data Common Variables (Continued)

6.1.4. Subroutine INPUT

This subroutine first computes the various noise generator amplitude gain constants stored in BLKDAT under GDB. These constants are calculated from corresponding input values for decibel signal to noise ratios. Next, the various parameters of the $\{\Gamma, \Phi, \Lambda\}$ matrices are calculated for the R, H, and J-filters. Then the equivalent noise bandwidths are calculated for R and J-filters. This sets the noise bandwidths of the corresponding multiplicative noise and jamming noise processes. Finally, the corresponding $\{\Gamma, \Phi, \Lambda\}$ parameters for the Kalman filter are calculated.

At the present time the $\{\Gamma, \Phi, \Lambda\}$ parameters for the Kalman filter are set equal to the true $\{\Gamma, \Phi, \Lambda\}$ used in the R, H, and J-filters. This is the "Perfect Identification" case. The theory supporting the above calculations is given below, in the description of Subroutine LPF.

NAME	TYPE	USES
NRHO	Integer *4	Order of I or Q state-vector for R-filter
NRHO2	Integer *4	Equal to twice NRHO
NH	Integer *4	Order of I or Q state-vector for H-filter
NH2	Integer *4	Equal to twice NH
NJ	Integer *4	Order of I or Q state-vector for J-filter
NJ2	Integer *4	Equal to twice NJ
NYRR	Integer *4	Kalman Filter mult. noise state-var. dim.
NYJR	Integer *4	Kalman Filter jamming state-var. dim.

Table 6.1.-4. INPUT Variables

6.1.5. Subroutine SIG

This subroutine produces sampled data waveforms at each sample number, KS, for the pseudo-random transmitted symbol in the alphabet {0, 1}. Symbol waveforms, BB, are produced for the direct path signal and waveforms, BBR, are produced for the differentially delayed reflected path.

NAME	TYPE	USES
KS	Integer *4	Sample number
BB	Integer *4	Symbol value, direct path
BBR	Integer *4	Symbol value, reflected path

Table 6.1.-5. SIG Variables

6.1.6. Subroutine MODUL

Subroutine MODUL accepts the outputs of Subroutine SIG, the zero-one symbol samples of BB and BBR, and converts them to the code symbols C and CR, respectively, in the {1, -1} alphabet, corresponding to C() of equation (3.2.-3). Next, the direct and reflected path sample times, TK and TR, corresponding to sample number KS are computed. Also, a modular time, TKMOD, referenced to the beginning of each new symbol is computed. Next, the envelope and phase functions, AT and PHEET, are computed, corresponding to equations (3.2.-4) and (3.2.-5). Finally, the I and Q modulation components of the transmitted signal are computed as XST(1) and XST(2), corresponding to equations (3.5.-9). Also computed are the differentially delayed I-Q components, FR and GR.

NAME	TYPE	USES
KS, BB, BBR	Same as Subroutine SIG	
XST	Real *4, 1-dimensional array, length 2	Direct-path, signal vector
FR	Real *4	Reflected-path I component
GR	Real *4	Reflected-path Q component

Table 6.1.-6. MODUL Variables

6.1.7. Subroutine LPF

At the present time, all of the colored I-Q noise processes are generated with no coupling between the in-phase and quadrature components. This yields band pass spectra which are even-symmetric with respect to the carrier frequency. Presently the I and Q scalar processes are obtained by filtering independent white scalar noise processes. The discrete-time filter routine which is presently used is a three-pole filter called LPF.

The discrete-time filter algorithms are obtained by assuming a continuous-time filter which is driven by a sampler and Zero-Order-Hold (ZOH). The output is then sampled synchronously with the input to produce a discrete-time filter. The reason that such an approach to discrete-time filtering is used is that for high sampling rates, the performance of the sampled system converges to that of the continuous system. Thus, all the engineering insight into continuous-time filtering may be directly applied to the discrete-time filtering.

Let the input waveform be $w(t)$ and the output waveform be $z(t)$. The operation of the filter is defined by

$$z(kT) = \underline{\lambda}^T \underline{x}(kT) \qquad : \quad k = 0, 1, 2, \ldots$$

$$\dot{\underline{x}}(t) = A\underline{x}(t) + \underline{b}m(t)$$

$$m(t) = w(kT) \qquad : \quad kT \leq t < (k+1)T \qquad (6.1.-1)$$

where T is the time interval between samples and k is the sample number. In (6.1.-1), $\underline{x}(t)$ is the filter state vector of dimension $N \times 1$. $\underline{\lambda}$ and $\underline{b}$ are also N-vectors. A is an $N \times N$ matrix. In particular

$$\underline{\lambda}^T = [\lambda_1, \lambda_2, \ldots, \lambda_N]$$

$$\underline{b}^T = [b_1, b_2, \ldots, b_N] \quad ,$$

$$A = \begin{bmatrix} p_1 & 0 & \cdots & 0 \\ 0 & p_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & p_N \end{bmatrix} \qquad (6.1\text{-}2)$$

Now it is well known that

$$G(s) = \frac{Z(s)}{M(s)} = \lambda^T (sI - A)^{-1} \underline{b}$$

$$= \frac{\lambda_1 b_1}{s - p_1} + \frac{\lambda_2 b_2}{s - p_2} + \frac{\lambda_3 b_3}{s - p_3} \quad : \quad N = 3 \qquad (6.1.-3)$$

For the diagonal A matrix, we see that the $\lambda_i b_i$ are the residues of G(s) at the poles $s = p_i$.

The equations (6.1.-1) may be solved for a difference equation in sample number k as

$$z(k) = \underline{\lambda}^T \underline{x}(k)$$

$$\underline{x}(k + 1) = \Phi \underline{x}(k) + \underline{\gamma} w(k)$$

$$\Phi = \exp(AT)$$

$$\underline{\gamma} = \int_0^T \exp(Aq)\, dq\, \underline{b} = (\Phi - I)\underline{b} \qquad (6.1.-4)$$

Solving (6.1.-4) for the diagonal A form yields

$$\Phi = \begin{bmatrix} e^{p_1 T} & 0 & \cdots & 0 \\ 0 & e^{p_2 T} & & \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & e^{p_N T} \end{bmatrix}$$

$$\underline{\gamma}^T = \left[\frac{b_1}{p_1}(e^{p_1 T} - 1), \frac{b_2}{p_2}(e^{p_2 T} - 1), \ldots, \frac{b_N}{p_N} e^{p_N T} - 1)\right] \qquad (6.1.-5)$$

For the present version of LPF, we simply choose a general formulation for G(s) as

$$G(s) = \frac{k(s - z)}{(s - p_1)(s - p_2)(s - p_3)}$$

$$p_i = -2\pi f_i \quad : \quad i = 1, 2, 3$$

$$z = -2\pi f_z \qquad (6.1.-6)$$

where the f_z and f_i are the 3-dB "break" frequencies in the corresponding Bode Plot. The 3 poles and single zero are all on the negative real axis of the S-plane. Under the assumption that the sampling rate is sufficiently high so that

$$M(s) \cong W(s) \quad : \quad \frac{Z(s)}{W(s)} \cong G(s) \qquad (6.1.-7)$$

then the behavior of the discrete-time filter of (6.1.-4) may be well characterized by simply choosing the break frequencies f_i, f_z and the gain constant, K.

When used to generate colored noise, it is desired that the output variance of LPF be unity when the input variance is unity. Then the desired variance is achieved by multiplying the output wave form by the gain constants, GJ or GRHO. When used to provide delay-spreading it is desired that LPF have unity low-frequency gain.

Unity low frequency gain for the discrete filter is obtained when the continuous time transfer function, G(s) has unity low-frequency gain. Thus, in (6.1.-6) the filter gain constant is solved for as

$$K = (2\pi)^2 \frac{f_1 f_2 f_3}{f_z} \quad : \quad \text{Unity Gain} \qquad (6.1.-8)$$

To set the filter for unity output variance requires analysis of the filter's transient variance. In general the output variance at sample k is

$$V_{zz}(k) = E\{z(k)z^T(k)\} = E\{(\underline{\lambda}^T \underline{x}(k))\ (\underline{\lambda}^T \underline{x}(k)^T\}$$

$$= \underline{\lambda}^T V_{xx}(k) \underline{\lambda} \qquad (6.1.-9)$$

where $V_{xx}(k)$ is the N x N variance matrix for the filter's state-vector. In general,

$$V_{xx}(k+1) = \Phi V_{xx}(k)\Phi^T + \underline{\gamma\gamma}^T \qquad (6.1.-10)$$

Let V_{xx} be the steady-state value of the variance matrix, $V_{xx}(k)$. Then, for the diagonal Φ-matrix, (6.1.-10) may be solved for V_{xx} as

$$V_{xx} = \left[\frac{\gamma_i\gamma_j}{1 - \phi_i\phi_j}\right] \quad : \quad i, j = 1, 2, \ldots, N \qquad (6.1.-11)$$

where i denotes column position and j denotes row position. Then the steady-state value of the output variance, V_{zz}, is

$$V_{zz} = \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\gamma_i\lambda_i\gamma_j\lambda_j}{1 - \phi_i\phi_j} \qquad (6.1.-12)$$

The G(s) of (6.1.-6) may be expanded in normal form as

$$G(s) = \sum_{i=1}^{3} \frac{r_i}{s - p_i}$$

$$r_1 = \frac{K(p_1 - z)}{(p_1 - p_2)(p_1 - p_3)}$$

$$r_2 = \frac{K(p_2 - z)}{(p_2 - p_1)(p_2 - p_3)}$$

$$r_3 = \frac{K(p_3 - z)}{(p_3 - p_1)(p_3 - p_2)} \qquad (6.1.-13)$$

From (6.1.-3), the residues, r_i, are

$$r_i = \lambda_i b_i \quad i = 1, 2, 3 \qquad (6.1.-14)$$

Since λ_i and b_i enter as a product, we may just set b_i to unity to get

$$\lambda_i = r_i \qquad (6.1.-15)$$

Thus, finally, we have

$$\phi_i = \exp(-2\pi f_i T)$$

$$\gamma_i = \frac{1 - \phi_i}{2\pi f_i}$$

$$\lambda_i = r_i$$

$$r_i = \frac{K(p_i - z)}{\prod_{\substack{i=1 \\ i \neq j}}^{3} (p_i - p_j)} \quad : \quad p_i = -2\pi f_i$$

$$K = (2\pi)^2 \frac{f_1 f_2 f_3}{f_z} \quad : \text{ Unity Gain}$$

$$K = \frac{2\pi}{\sqrt{\sum_{i=1}^{3} \sum_{j=1}^{3} \frac{(1 - \phi_i)(1 - \phi_j) q_i q_j}{f_i f_j (1 - \phi_i \phi_j)}}} \quad : \text{ Unity Variance} \qquad (6.1.-16)$$

$$q_i = \frac{r_i}{K}$$

Equations (6.1.-4) and (6.1.-16) define the operation of the discrete-time filter, LPF.

It is also desired to compute the one-sided equivalent noise bandwidth, B_N, for the filter LPF. This is done from the formulation

$$B_N = \frac{1}{2G^2(0)} \sum\{\text{Residues of } G(s)G(-s) \text{ in the Left Half S-plane}\} \qquad (6.1.-17)$$

After much arithmetic manipulation it can be shown that B_N is given by

$$B_N = (\frac{\pi}{2})(\frac{f_1 f_2 f_3}{f_z^2}) \left[\frac{(f_1^2 - f_z^2)(f_2^2 - f_3^2)(f_2 f_3) + (f_2^2 - f_z^2)(f_3^2 - f_1^2)(f_1 f_3)}{(f_1^2 - f_2^2)(f_2^2 - f_3^2)(f_3^2 - f_1^2)} \right.$$

$$\left. + \frac{(f_3^2 - f_z^2)(f_1^2 - f_2^2)(f_1 f_2)}{} \right] \qquad (6.1.-18)$$

Subroutine LPF simply implements equation (6.1.-1) using values for $\{\underline{\gamma}, \Phi, \underline{\lambda}\}$ which are computed in subroutine INPUT. The arithmatic in LPF is done in vector-matrix form using a special package of vector-matrix subroutines written for this simulation. The subroutine LPF is actually structured to do the indicated filtering twice, once on an in-phase input and once again on a quadrature input. Thus, LPF processes a two-vector and produces a two-vector.

NAME	TYPE	USE
KS	Integer*4	Sample number
N	Integer*4	Dimen. of I or Q state-vector
N2	Integer*4	Twice N
W	Real*4, 1 dim. array, length 2	Input vector
GI	Real*4, 1 dim. array, length 5	$\underline{\gamma}$ components
PI	Real*4, 1 dim. array, length 5	Φ diagonal components
CI	Real*4, 1 dim. array, length 5	$\underline{\lambda}$ components
XK	Real*4, 1 dim. array, length 10	state vector, dim. 2N
YK	Real*4, 1 dim. array, length 2	output vector

Table 6.1.-7. LPF Variables

6.1.8. Subroutine RHOFILT

This subroutine generates the low-pass multiplicative noise two-vector, $\underline{\rho}(k)$, of equation (3.5.-10). At the present time, $\underline{\rho}(k)$ is generated for the purely diffuse reflection case, without specular component. Also, the R.F. spectrum resulting from $\underline{\rho}(k)$ is taken as even symmetric with respect to the carrier frequency. Thus, the I and Q components of $\underline{\rho}(k)$ are uncorrelated. RHOFLT simply calls MARSA twice to obtain two independent, white, Gaussian, zero-mean, unit-variance noise samples. Then LPF is called to filter the noise samples. At each iteration the state-vector, XKRHO is stored so as to keep the filter running from sample to sample. The produced unit-variance output vector RHO is multiplied by gain constant GRHO to set the final desired multiplicative noise variance.

NAME	TYPE	USE
KS	Integer *4	Sample number
NRHO	Integer *4	Dimension of I or Q Vector
NRHO2	Integer *4	Twice NRHO
XKRHO	Real *4, 1-dimensional array, length 10	Mult. noise state vector
RHO	Real *4, 1-dimensional array, length 2	Mult. noise vector

Table 6.1.-8. RHOFLT Variables

6.1.9. Subroutine HFILTR

This subroutine provides delay-spreading filtering for the diffentially-delayed, reflected signal. This filter corresponds to the h(k) of equation (3.5.-10). The inputs to HFILTR and the I and Q components, FR and FR, created by MODUL. HFILTR calls LPF to filter these inputs to create an output two-vector FHGH. The HFILTR may be turned on or off according to the option NOH equals 0 or 1. For NOH = 0, the filter is employed. For NOH = 1, the input is hard-wired directly through to the output.

NAME	TYPE	USE
KS	Integer *4	Sample number
NH	Integer *4	Dimension of I or Q vector
NH2	Integer *4	Twice NH
FR	Real *4	Input I component
GR	Real *4	Input Q component
XKH	Real *4, 1-dimensional array, length 10	Filter state-vector
FHGH	Real *4, 1-dimensional array, length	Output I-Q vector

Table 6.1.-9. HFILTR Variables

6.1.-10. Subroutine GEN

This subroutine combines the output of HFILTR and RHOFLT to produce the doubly-spread reflected signal. From the two-vector, FHGH, is produced the rotation matrix, $H_t()$, of equation (3.5.-10) which is called SOOK. The multiplicative noise vector, RHO, is then multiplied by SOOK to produce the doubly-spread I-Q vector, XSR corresponding to $\underline{y}'_r(k; m)$ of equation (3.5.-10). The multiplication is done by calling MATVEC which is a special routine for doing matrix times vector multiplication.

NAME	TYPE	USE
RHO	Real *4, 1-dimensional array, length 2	Mult. noise I-Q vector
FHGH	Real *4, 1-dimensional array, length 2	Delay-spread I-Q vector
XSR	Real *4, 1-dimensional array, length 2	Doubly-spread I-Q vector

Table 6.1.-10. GEN Variables.

6.1.-11. Subroutine INT2

This subroutine produces the stochastic jamming signal and is one-half of what will eventually be the total generator for deterministic or stochastic jamming, called INT. This subroutine, INT2 calls MARSA twice to obtain two independent, white, zero-mean, unit-variance, Gaussian noise samples. Then LPF is called to filter these samples to produce $\underline{\rho}_j(k)$ of equation (3.5.-22) as YJ. Next, the rotation matrix, $H_j(k)$, of (3.5.-18) is formed as HJ. This is the rotation due to a possible frequency offset of the jamming signal from the desired signal. Next, $H_j(k)$ is multiplied by $\underline{\rho}_j(k)$ to produce the output I-Q vector, XJ, which corresponds to $\underline{y}'_J(k)$ in (3.5.-17). The output XJ is given the correct variance by multiplying by the gain constant, GJ.

NAME	TYPE	USE
KS	Integer*4	Sample number
NJ	Integer*4	Dimension of I or Q jamming component
NJ2	Integer*4	Twice NJ
XKJ	Real*4, 1-dim. array, length 10	Jamming noise state-vector
XJ	Real*4m 1-dim. array, length 2	Jamming noise I-Q vector

Table 6.1.-11. INT2 variables

6.1.-12. Subroutine NOISE

This subroutine forms the I-Q two-vector which is the zero-mean, white Gaussian additive channel noise. The output is XN, which corresponds to $\underline{n}(k)$ of (3.5.011). This routine simply calls MARSA twice and then muliplies the result by the variance-setting gain constant, GN. The only variable which is passed is XN, the output I-Q vector which is Real*4, a one-dimensional array of length 2.

6.1.-13. Note on setting the Gain Constants

In generating the stochastic colored multiplicative noise for multipath, colored additive noise for jamming, and white additive noise for the receiver, three gain functions are used to set the variances of the stochastic process generators. It is convenient to set these variances according to the various signal to interference ratios which exist at the R.F.

From Chapter Three, the various continuous-time signals and interferences may be written as

$$s_t(t) = A_t(t)\cos[\omega_c t + \phi_t(t)] \quad : \quad \text{Direct-path signal}$$

$$s_r(t) = \rho_i(t)\, A_t(t)\cos[\omega_c t + \phi_t(t)] - \rho_q(t)\, A_t(t)\sin[\omega_c t + \phi_t(t)];$$

Reflected Signal

$$s_j(t) = \rho_{ji}(t)\cos[(\omega_c + \Delta\omega_j)t] - \rho_{jq}(t)\sin[(\omega_c + \Delta\omega_j)t];$$

Jamming Process

$$n(t) = n_i(t)\cos\omega_c t - n_q(t)\sin\omega_c t \; ; \quad \text{White Noise} \qquad (6.1.\text{-}19)$$

For constant-envelope signals (PSK, FSK, DPSK)

$$A_t(t) = 1 \qquad (6.1.\text{-}20)$$

and the direct-path signal power is

$$S = E\{s_t^2(t)\} = 1/2 \qquad (6.1.-21)$$

For the completely diffuse reflection, without modulation ($\phi_t(t) = 0$), the reflected-path autocorrelation function is

$$R_{rr}(\tau) = R_{ii}(\tau) \cos \omega_c\tau + R_{iq}(\tau) \sin \omega_c\tau$$

$$R_{ii}(\tau) = E\{\rho_i(t + \tau)\rho_i(t)\} \quad : \text{ even function}$$

$$R_{iq}(\tau) = E\{\rho_i(t + \tau)\rho_q(t) \quad : \text{ odd function} \qquad (6.1.-22)$$

under the condition that $\rho_i(t) \cos \omega_c t - \rho_q(t) \sin \omega_c t$ is a weakly stationary process. *The reflected signal power is then*

$$M = R_{rr}(0) = R_{ii}(0) = E\{\rho_i^2(t)\} = E\{\rho_q^2(t)\} = \sigma_\rho^2 \qquad (6.1.-23)$$

Likewise, the jamming power and white noise power are

$$J = E\{\rho_{ji}^2(t)\} = E\{\rho_{jq}^2(t)\} = \sigma_j^2$$

$$N = E\{n_i^2(t)\} = E\{n_q^2(t)\} = \sigma_n^2 \qquad (6.1.-24)$$

Thus, signal to noise power ratios may be defined as

$$\frac{S}{N} = \frac{1}{2\sigma_n^2} = \text{SNR} \quad ; \quad \text{Direct-path to White Noise}$$

$$\frac{S}{J} = \frac{1}{2\sigma_j^2} = \text{SJR} \quad ; \quad \text{Direct-path to Jamming}$$

$$\frac{S}{M} = \frac{1}{2\sigma_\rho^2} \quad ; \quad \text{Direct-path to Reflected Path} \qquad (6.1.-25)$$

Actually, the simulation uses the "Multipath Ratio,"

$$MPR = \frac{M}{S} = 2\sigma_\rho^2 \qquad (6.1.-26)$$

Now, because of the manner in which the various stochastic processes are generated, the gain constants are defined by

$$G_n^2 = \sigma_n^2$$

$$G_j^2 = \sigma_j^2$$

$$G_\rho^2 = \sigma_\rho^2 \qquad (6.1.-26)$$

Thus

$$G_n = \frac{1}{\sqrt{2SNR}} = \frac{1}{\sqrt{2}} \cdot 10^{-\left(\frac{SNRDB}{20}\right)} = GN$$

$$G_j = \frac{1}{\sqrt{2SJR}} = \frac{1}{\sqrt{2}} \cdot 10^{-\left(\frac{SJRDB}{20}\right)} = GJ$$

$$G = \sqrt{\frac{MPR}{2}} = \frac{1}{\sqrt{2}} \cdot 10^{+\left(\frac{MPRDB}{20}\right)} = GRHO \qquad (6.1.-27)$$

In the case of the white additive noise, it is more convenient for graph-plotting purposes to set GN according to the ratio of symbol energy divided by noise spectral density, E/N_o. The E/N_o ratio is computed only for that energy actually used for signalling, and does not include any energy in residual unmodulated carrier components. Thus, in general, E/N_o and GN must be computed separately for each modulation case. From (4.4.-23) and (4.3.-29) result

$$GN = \sqrt{\frac{K}{2}} \cdot |\sin(\Delta\phi)| \cdot 10^{-\left(\frac{ENODB}{20}\right)} \quad ; \text{PSK}$$

(continued)

$$GN = \sqrt{\sum_{k=1}^{K} \sin^2\left(\frac{\Delta\omega T}{K}(k - 1/2)\right)} \cdot 10^{-\left(\frac{ENODB}{20}\right)} \quad ; \text{FSK} \qquad (6.1.-28)$$

6.1.14. Subroutine DATA

This subroutine simply adds the I-Q vector outputs of MODUL, GEN, INT2 and NOISE to produce the received data vector Z. Z is real*4, a one-dimensional array of length 2.

6.1.15. Subroutine REFGEN

This subroutine generates the signal references for the linear filter. In particular, it generates the references for the transmitted and reflected components XST, FR, and GR, respectively. Each reference signal FTR, GTR, FRR, and GRR, is generated twice, once for a transmitted symbol, m = 0, and once for a transmitted symbol, m = 1. Each time REFGEN is called the desired symbol, m, is specified. The time reference, TKMOD, is time since beginning of the current symbol. This is the same time reference used in MODUL. The direct-path signal references, FTR and GTR, have a gain constant AEST, representive identification of direct-path signal strength.

NAME	TYPE	USE
KS	Integer*4	Sample number
M	Integer*4	Symbol value
FTR	Real*4	Direct-path signal I reference
GTR	Real*4	Direct-path signal Q reference
FRR	Real*4	Refl.-path signal I reference
GRR	Real*4	Refl.-path signal Q reference

Table 6.1.-12 REFGEN Variables

6.1.-16. Subroutine KALFLT.

This is the subroutine which actually performs the linear filtering and tracking of the colored channel interferences in the presence of white noise. The diagram of Figure 4.2.-1 shows that the filter may be separated into two parallel paths coupled only through the Innovations process, one path for jammer-tracking and one path for multipath-tracking. However, as noted in Section 4.2., the Kalman gain equations are coupled and cannot be split into two parallel computations. Thus, the present version of KALFLT employs a state-vector which is a partitioned vector made up of the jamming and multipath state vectors. Thus, for the case where the I and Q component state vectors for jamming and multipath were all of dimension 3, the partitioned state vector for KALFLT is of dimension 12. The error variance matrices in the gain computation are 12 x 12.

The first steps in KALFLT are to set up the structure of the filter in terms of identified values for the $\{\Gamma, \Phi, \Lambda\}$ matrices for both the jamming and multipath processes. These correspond to the $\{\Gamma_r, \Phi_r, \Lambda_r\}$ and $\{\Gamma_j, \Phi_j, \Lambda_j\}$ of equation (4.3.-2). Also, the signal reference matrices, $H_t()$ and $H_j()$, are set up using the I-Q components FTR, GRT, FRR, GRR provided by REFGEN as well as the identified estimates of the parameters, GN, GRHO, GJ, and others.

After the setting up of the Kalman filter structure, the filter equations, themselves, are implemented. These are the partitioned version of (4.2.-2) plus the partitioned gain equations. These equations are presently for the case of colored multiplicative and additive noises plus white noise. The equations for delay spreading have not been implemented.

The reason that the setting up of the Kalman filter structure has been included in KALFLT, rather than in INPUT, is to cover the possibility that the multipath or jamming processes are (slowly) non-stationary. In that case the Identification algorithms would be continuously adjusting the Kalman filter structure. Under an assumption of stationarity or for a fixed filter configuration (sub-optimal), the setting up of the filter structure could be done in INPUT, thus speeding up the execution of KALFLT. An even greater increase in execution speed can be obtained by using the

steady-state (Wiener) version of the Kalman filter. Then the gain equations (12 x 12) themselves may be moved to INPUT, leaving only the filter equations in KALFLT. This modification is presently being pursued.

It can be seen from the KALFLT listing (Section 6.2) that both the gain and filter equations are executed in vector-matrix arithmetic. The applicable subroutines are described below.

NAME	TYPE	USE
KS	Integer*4	Sample number
NYRR	Integer*4	Mul. noise state-vector dimension
NYJR	Integer*4	Jamming noise state-vector dimension
FTR, GTR	Real*4	Direct-path I-Q signal references
FRR, GRR	Real*4	Refl.-path I-Q signal references
Z	Real*4, 1-dim. array, length 2	Input data I-Q vector
XEST	Real*4, 1-dim. array, length 12	Filtered state-vector
VEST	Real*4, 2-dim. array, 12 x 12	Filtered error covariance matrix
AINNOV	Real*4, 1-dim. array, length 2	Innovations vector
VINV	Real*4, 2-dim. array, 2 x 2	Innovations variance inverse matrix
DET	Real*4	Determinant of Innovations variance matrix

Table 6.1.-13. KALFLT Variables

6.1.-17. Subroutine DCIDM

This is the subroutine which forms the IDEI decision statistics, S_i, of equation (4.1.-12) using the arguments of the Gaussian density of equation (4.1.-13). The inputs are the Innovations process, its inverse covariance matrix, and the determinant of its covariance matrix. Instead of comparing the products of (4.1.-12) at the Kth (NSPB) sample time, the natural logarithms of the products are compared. Since the $\log_e()$ function is monotonic, non-decreasing, and positive, this also gives a sufficient statistic. Since the desity is Gaussian, the cumulative product is computed as a cumulative sum of the logarithms.

NAME	TYPE	USE
KS	Integer*4	Sample number
AINNOV	Real*4, 1-dim. array, length 2	Innovations vector
VINV	Real*4, 2-dim. array, 2 x 2	Innovations variance inverse matrix
DET	Real*4	Determinant, Innov. variance
SUM	Real*4	Cumulative statistic value
S	Real*4	Final statistic value

Table 6.1.-14. DCIDM Variables

6.1.-18. Subroutine STDCIM

This is the subroutine which computes the decision statistics for the standard PSK coherent detector and FSK incoherent detector as in equations (4.3.-30) and (4.3.-41), respectively. The PSK test of (4.3.-30) is equivalent to simply summing the Q data samples, Z(2), over the symbol period and then testing for positiveness or negativeness. This equivalent test is actually done. Thus, the PSK statistic is simply the sum of the Z(2) data samples. For FSK, the exact operation of (4.3.-41) is implemented.

NAME	TYPE	USE
KS	Integer*4	Sample number
Z	Real*4, 1-dim array, length 2	Data vector
SUM	Real*4	Cumulative PSK statistic
S	Real*4	Final PSK statistic
FTR0, GRT0, FTR1, GTR1	Real*4	Signal reference waveforms
AFSK0, AFSK1, BFSK0, BFSK1	Real*4	a(k:0), a(k:1), b(k:0), b(k:1)
SFSK0 - SFSK1	Real*4	LR(K) in equation (4.3.-41)

Table 6.1.-15. STDCIM Variables

6.1.19. Subroutine CERKAL

This subroutine is called by the main program after the last data sample has been processed in each symbol interval. The decision statistics, S0, and S1, produced by DC1DM are compared to make the symbol decision. The true symbol is BB. The detected symbol is BBHAT. Based on the symbol decision, the final state estimates, XEST, in both Kalman filters are set equal to the estimate in the "correct" filter. Also, the estimation error variance matrices are symmetrized and set equal. This latter step is to combat the cumulative effects of round-off error in the 12 x 12 matrix computations. If BB does not equal BBHAT, an error counter is incremented. A cumulative raw error rate, ERRATE, is also computed. Finally, subroutine ESTERR is called to compute smoothed mean error rate and error rate variance from the raw error rate.

NAME	TYPE	USE
KS	Integer*4	Sample number
NYRR, NYJR	Integer*4	Dim. of Jam. and Mult. state-vectors
SO, S1	Real*4	Dec. Stx. from DC1DM
XESTO, XEST1	Real*4, 1 dim. array, length 12	Kal. filter state vectors
VESTO, VEST1	Real*4, 2-dim. arrays, 12 x 12	Kal. filter error var. matrices
BB	Integer*4	Transmitted symbol, 0 or 1
ERR	Real*4	Cumulative symbol error count
ERRATE	Real*4	Raw symbol error rate
DIB	Real*4	Smoothing constant
SMEAN	Real*4	Smoothed error rate mean, unnormalized
ERMEAN	Real*4	Smoothed error rate mean
SVAR	Real*4	Smoothed error rate variance, unnormalized
ERRVAR	Real*4	Smoothed error rate variance
DATSNR	Real*4	Ratio of square of smoothed error rate mean to smoothed error rate variance

Table 6.1.-16. CERKAL Variables

6.1.20. Subroutine CERSTD

This is the counterpart of CERKAL which is used to compute error rates and variances for the standard detector. The operation and variable list is completely analagous to that for CERKAL.

6.1.21. Subroutine ESTERR

This subroutine smooths the raw ERRATE to generate a mean error rate for either CERKAL or CERSTD. Also, ESTERR generates the smoothed error rate variance and data signal to noise ratio, DATSNR. This routine also tests the simulation results to terminate the simulation run. Three tests must be met simultaneously to terminate. First, the computed DATSNR must be greater than a set-in value, DSRLMT. Second, the number of symbols processed must be greater than a set-in limit, NIB. Third, the smoothed error-rate mean must satisfy a Cauchy convergence criterion. This criterion is that the absolute difference of the two most recent ERMEAN values, divided by the most recent, must be less than a set-in value, CAVAL. When the errors occur in bursts, it is generally this Cauchy criterion which controls the simulation termination. The ESTERR variables are described in Table 6.1.-16, for CERKAL.

6.1.22. Subroutine MARSA

This subroutine generates independent Gaussian numbers having zero mean and unit variance. It is due to Marsaglia [105] and is faster than the Box-Muller [106] method. Input to the subroutine are pairs of independent numbers which are uniformly distributed on the interval, [-1, 1]. These numbers are obtained from a version of the IBM-360 library routine, RANC, which has been rewritten for the PDP-11/40. For remote simulation on the CDC-6600, its library routine, RANF, is used in lieu of RANC. The CDC routine requires two other subroutines, RANSET and RANGET, which supply seed numbers to RANF. The subroutine MARSA is simple and self-explanatory.

6.1.23. The Matrix-Vector Subroutines

Rather than rely on unknown matrix-vector algebra library subroutines on host machines, the few routines needed were written specifically for this simulation. These routines are simple and self-explanatory from their listings in Section 6.2. They are

MATMUL	:	Matrix multiplication
MATVEC	:	Matrix times vector
VECAS	:	Vector addition (or subtraction)
MATAS	:	Matrix addition (or subtraction)
MATINV	:	Matrix inverse

6.2. PROGRAM LISTINGS

```
C     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C     THIS IS CALLED THE YOONM4 WHICH IS THE MAIN SIMULATION
C     WITHOUT STEADY-STATE KALMAN FILTER
C     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      REAL INNOV0,INNOV1
      DIMENSION XKH(10),XKRHO(10),XKJ(10)
      DIMENSION FHGH(2),RHO(2),XST(2),XSR(2),XN(2),XJ(2),Z(2)
      DIMENSION XEST0(12),XEST1(12),VEST0(12,12),VEST1(12,12),
     $INNOV0(2),INNOV1(2),VINV0(2,2),VINV1(2,2)

C
      COMMON/ORDER/NRHO,NRHO2,NH,NH2,NJ,NJ2,NYRR,NYJR
      COMMON/CRTN/NDS,NPRNT,NIB,NCASEQ
      COMMON/SAMPLE/NSPB,TB,TBR
      COMMON/RATIO/RHODB,SNRDB,SJRDB,RHODBR,SNRR,SJRDBR

C
      LOGICAL*1 STRNG(8)
      INTEGER*4 JTIME
      CALL ASSIGN(4,'TT:')
      CALL ASSIGN(5,'CR:')
      CALL ASSIGN(6,'KB:/C')

C
      CALL INPUT

C
C  MAIN
C
      WRITE(6,65)
   65 FORMAT(9X,10HERR IN KAL,4X,9HERR IN ST,5X,12HERATE IN KAL,
     12X,11HERATE IN ST,3X,13HERMEAN IN KAL,1X,12HERMEAN IN ST,2X,
     212HERVAR IN KAL,2X,11HERVAR IN ST)
      CALL GTIM(JTIME)
      CALL TIMASC(JTIME,STRNG)
      WRITE(6,7272) (STRNG(II),II=1,8)
 7272 FORMAT(1X,'START TIME IS ',8A1)
C
      DO 1000 KS=1,NDS
      CALL SIG(KS,BB,BBR)
      CALL MODUL(KS,BB,BBR,XST,FR,GR)
      CALL RHOFLT(KS,NRHO,NRHO2,XKRHO,RHO)
      CALL HFILTR(KS,NH,NH2,FR,GR,XKH,FHGH)
      CALL GEN(RHO,FHGH,XSR)
      CALL NOISE(XN)
      CALL INT2(KS,NJ,NJ2,XKJ,XJ)
      CALL DATA(XST,XSR,XJ,XN,Z)
      CALL REFGEN(KS,0,FTR0,GTR0,FRR0,GRR0)
      CALL REFGEN(KS,1,FTR1,GTR1,FRR1,GRR1)
      CALL KALFLT(KS,NYRR,NYJR,FTR0,GTR0,FRR0,GRR0,Z,XEST0,VEST0,
     $INNOV0,VINV0,DET0)
      CALL KALFLT(KS,NYRR,NYJR,FTR1,GTR1,FRR1,GRR1,Z,XEST1,VEST1,
     $INNOV1,VINV1,DET1)
      CALL DCIDM(KS,INNOV0,VINV0,DET0,SUM0,S0)
      CALL DCIDM(KS,INNOV1,VINV1,DET1,SUM1,S1)
      CALL STDCIM(KS,Z,SUMS,SS,FTR0,GTR0,FTR1,GTR1,AFSK0,
     $AFSK1,BFSK0,BFSK1,SFSK0,SFSK1)

C
      IB=1+IFIX((KS-0.5)/NSPB)

C
      IF(MOD(KS,NSPB).NE.0) GO TO 1000
      CALL CERKAL(KS,NYRR,NYJR,S0,S1,XEST0,XEST1,VEST0,VEST1,
     $BB,ERR,ERRATE,DIB,SMEAN,ERMEAN,SVAR,ERRVAR,DATSNR)
      CALL CERSTD(KS,SS,SFSK0,SFSK1,BB,ERRS,ERATES,DIBS,SMEANS,
     1ERMENS,SVARS,ERVARS,DATSNS)
C
```

```
      IF(MOD(IB,NPRNT).NE.0) GO TO 1000
      CALL GTIM(JTIME)
      CALL TIMASC(JTIME,STRNG)
      WRITE(6,7273) (STRNG(II),II=1,8)
7273  FORMAT(1X,'TIME IS ',8A1)
      WRITE(6,66) IB, ERR,ERRS,ERRATE,ERATES,ERMEAN,ERMENS,ERRVAR,
     $ERVARS
   66 FORMAT(1X,'IB=',I3,8E14.6)
 1000 CONTINUE
      WRITE(6,200) SNRDB,RHODB,SJRDB,NSFB,IB,ERMEAN,ERRVAR,DATSNR
  200 FORMAT(1H0,2X,'PRIMARY DATA OUTPUTS:',5X,
     1'CASE NUMBER=',3(5X,E13.6),/,5X,
     2'NUMBER OF DATA SAMPLES=',I10,/,5X,
     3'NUMBER OF MESSAGE SYMBOLS=',I5,/,5X,
     4'MEAN ERROR RATE=',E13.6,/,5X,
     5'ERROR RATE VARIANCE=',E13.6,/,5X,
     6'DATA SNR=',E13.6)
      STOP
      END
```

```
      BLOCK DATA
C     DECLARATION INFORMATION
      INTEGER*2 IX1(2),JX1(2)              ! USED IN SUBROUTINE SIG
      INTEGER*2 IX2(2),JX2(2),IX3(2),JX3(2) ! USED IN SUBROUTINE RHOFLT
      INTEGER*2 IX4(2),JX4(2),IX5(2),JX5(2) ! USED IN SUBROUTINE NOISE
      INTEGER*2 IX6(2),JX6(2),IX7(2),JX7(2) ! USED IN SUBROUTINE INT2
C
      INTEGER*4 IXS,JXS                    ! USED IN SUBROUTINE SIG
      INTEGER*4 IXG1,JXG1,IXG2,JXG2        ! USED IN SUBROUTINE RHOFLT
      INTEGER*4 IXN1,JXN1,IXN2,JXN2        ! USED IN SUBROUTINE NOISE
      INTEGER*4 IXJ1,JXJ1,IXJ2,JXJ2        ! USED IN SUBROUTINE INT2
C
      EQUIVALENCE (IXS,IX1),(JXS,JX1) ! FOR SIG
      EQUIVALENCE (IXG1,IX2),(JXG1,JX2),(IXG2,IX3),(JXG2,JX3) !FOR RHOFLT
      EQUIVALENCE (IXN1,IX4),(JXN1,JX4),(IXN2,IX5),(JXN2,JX5) !FOR NOISE
      EQUIVALENCE (IXJ1,IX6),(JXJ1,JX6),(IXJ2,IX7),(JXJ2,JX7) ! FOR INT2
C
      DATA IX1,JX1/"136303,"053354,"041256,"141560/ ! FOR SIG
      DATA IX2,JX2,IX3,JX3/"176303,"037702,"141236,"056407,
     1                     "125537,"103453,"055052,"032461/ ! FOR RHOFLT
      DATA IX4,JX4,IX5,JX5/"034313,"103400,"021165,"104262,
     1                     "072063,"122076,"016415,"041540/ ! FOR NOISE
      DATA IX6,JX6,IX7,JX7/"076423,"102132,"116055,"123556,
     1                     "071743,"023100,"104165,"021331/ ! FOR INT2
C
      COMMON /SEED/ IXS,JXS,IXG1,JXG1,IXG2,JXG2,IXN1,JXN1,IXN2,JXN2,
     1              IXJ1,JXJ1,IXJ2,JXJ2
C
C     FOR CALCULATION THE CONSTANT COEFFICIENT MATRIX OF SUBROUTINE
C     LPF(LOW PASS FILTER) FOR SUB.RHOFLT,SUB.HFILTR AND SUB.INT2
C
      REAL GAMH(5),PHEEH(5),CH(5),GAMRHO(5),PHERHO(5),CRHO(5),
     1GAMJ(5),PHEEJ(5),CJ(5)
      COMMON /ZPWUP/ GAMH,PHEEH,CH,GAMRHO,PHERHO,CRHO,GAMJ,PHEEJ,CJ
C
C     FOR CALCULATION THE CONSTANT COEFFICIENT MATRIX OF KALMAN FILTER
C
      REAL GAMRR(6,2),PHEERR(6,6),CRR(2,6),GAMJR(6,2),PHEEJR(6,6),
     $CJR(2,6),MUR(2),MUJ(2)
      COMMON/ZPWKAL/GAMRR,PHEERR,CRR,GAMJR,PHEEJR,CJR,MUR,MUJ
C
      COMMON/ORDER/NRHO,NRHO2,NH,NH2,NJ,NJ2,NYRR,NYJR
      COMMON/DATASN/DSRLMT,ALFA
      COMMON/CRTN/NDS,NPRNT,NIB,NCASEQ
      COMMON/SAMPLE/NSPB,TB,TBR
      COMMON/DELAY/DEL,DELPHI,DELMEG,DELMEJ,DELJR
      COMMON/AMAG/AT,AR,AEST
      COMMON/OPTION/NOH,NOS
      COMMON/RATIO/ENODB,RHODB,SJRDB,ENODBR,RHODBR,SJRDBR
      COMMON/GDB/GRHO,GN,GJ,GRHOR,GNR,GJR
C
      COMMON /SIGCON/ SIGB(50)
      DATA SIGB/50*0.0/
C
      END
```

```
C
      SUBROUTINE INPUT
      COMMON/ORDER/NRHO,NRHO2,NH,NH2,NJ,NJ2,NYRR,NYJR
      COMMON/DATASN/DSRLMT,ALFA
      COMMON/CRTN/NDS,NPRNT,NIB,NCASEQ
      COMMON/SAMPLE/NSPB,TB,TBR
      COMMON/DELAY/DEL,DELPHI,DELMEG,DELMEJ,DELJR
      COMMON/RATIO/ENODB,RHODB,SJRDB,ENODBR,RHODBR,SJRDBR
      COMMON/OPTION/NOH,NDS
      COMMON/AMAO/AT,AR,AEST
      COMMON/GDB/GRHO,GN,GJ,GRHOR,GNR,GJR
      COMMON/ZPWUP/GAMH,PHEEH,CH,GAMRHO,PHERHO,CRHO,GAMJ,PHEEJ,CJ
      COMMON/ZPWKAL/GAMRR,PHEERR,CRR,GAMJR,PHEEJR,CJR,MUR,MUJ
      DIMENSION GAMH(5),PHEEH(5),CH(5),GAMRHO(5),PHERHO(5),CRHO(5),
     $GAMJ(5),PHEEJ(5),CJ(5)
      DIMENSION GAMRR(6,2),PHEERR(6,6),CRR(2,6),GAMJR(6,2),PHEEJR(6,6),
     1CJR(2,6),MUR(2),MUJ(2)
      REAL MUR,MUJ
      DIMENSION FRHO(5),FH(5),FJ(5),RRHO(5),RH(5),RJ(5)
C
      LOGICAL*1 INTQ,YES
      DATA YES/'Y'/
C
      READ(5,1) NRHO,NRHO2,NH,NH2,NJ,NJ2,NYRR,NYJR
      READ(5,2) NSPB,TB,TBR
      READ(5,3) NOH,NDS
      READ(5,4) DEL,DELPHI,DELMEJ,DELJR
      READ(5,5) AT,AR,AEST
      READ(5,5) ENODB,RHODB,SJRDB
      READ(5,5) ENODBR,RHODBR,SJRDBR
      READ(5,6) LSEQ,NPRNT,NIB,NCASEQ
      READ(5,7) DSRLMT,ALFA
      READ(5,20) FZRHO,(FRHO(I),I=1,3)
      READ(5,20) FZH,(FH(I),I=1,3)
      READ(5,20) FZJ,(FJ(I),I=1,3)
    1 FORMAT(8I5)
    2 FORMAT(I5,2E15.6)
    3 FORMAT(2I5)
    4 FORMAT(4E15.6)
    5 FORMAT(3E15.6)
    6 FORMAT(4I10)
    7 FORMAT(2E15.6)
   20 FORMAT(4E15.6)
      NDS=LSEQ*NSPB
      PI=4.*ATAN(1.0)
      DELMEG=DELPHI*2.*PI/TB
      IF(NDS.EQ.1) GO TO 30
      SUMF=0.0
      DO 35 K=1,NSPB
   35 SUMF=SUMF+SQRT(SIN((K-0.5)*TB*DELMEG/NSPB)**2)
      GN=SUMF*10.**(-ENODB/20.)
      GNR=SUMF*10.**(-ENODBR/20.)
      GO TO 40
   30 CONSTP=SQRT(NSPB/2.)*ABS(SIN(DELPHI))
      GN=CONSTP*10.**(-ENODB/20.)
      GNR=CONSTP*10.**(-ENODBR/20.)
   40 CONTINUE
      GRHO=10.**(RHODB/20.)/SQRT(2.)
      GJ=10.**(-SJRDB/20.)/SQRT(2.)
      GRHOR=10.**(RHODBR/20.)/SQRT(2.)
```

```
      GJR=10.**(-SJRDBR/20.)/SQRT(2.)
      T=TB/NSPB
      DO 200 L=1,3
      PHEEH(L)=EXP(-2.*PI*FH(L)*T)
      PHERHO(L)=EXP(-2.*PI*FRHO(L)*T)
      PHEEJ(L)=EXP(-2.*PI*FJ(L)*T)
  200 CONTINUE
      DO 210 I=1,3
      R1=1.
      R2=1.
      R3=1.
      DO 215 J=1,3
      IF(J.EQ.I) GO TO 215
      R1=R1*(FRHO(J)-FRHO(I))
      R2=R2*(FH(J)-FH(I))
      R3=R3*(FJ(J)-FJ(I))
  215 CONTINUE
      RRHO(I)=(FZRHO-FRHO(I))/(2.*PI*R1)
      RH(I)=(FZH-FH(I))/(2.*PI*R2)
      RJ(I)=(FZJ-FJ(I))/(2.*PI*R3)
  210 CONTINUE
      DO 230 K =1,3
      GAMRHO(K)=(1.-PHERHO(K))/(2.*PI*FRHO(K))
      GAMH(K)=(1.-PHEEH(K))/(2.*PI*FH(K))
      GAMJ(K)=(1.-PHEEJ(K))/(2.*PI*FJ(K))
  230 CONTINUE
      HK=(2.*PI)**2.*FH(1)*FH(2)*FH(3)/FZH
      RHOK=0.0
      FJK=0.0
      DO 250 I=1,3
      DO 250 J=1,3
      RHOK=RHOK+RRHO(I)*GAMRHO(I)*RRHO(J)*GAMRHO(J)
     $/(1.-PHERHO(I)*PHERHO(J))
      FJK=FJK+RJ(I)*GAMJ(I)*RJ(J)*GAMJ(J)/(1.-PHEEJ(I)*PHEEJ(J))
  250 CONTINUE
      RHOK=1./SQRT(RHOK)
      FJK=1./SQRT(FJK)
      DO 70 II=1,3
      CRHO(II)=RHOK*RRHO(II)
      CH(II)=HK*RH(II)
      CJ(II)=FJK*RJ(II)
   70 CONTINUE
C
C     ONE-SIDED EQUIVALENCE NOISE BANDWIDTH FOR THE MULTIPLICATIVE
C     AND THE JAMMING PROCESS
C
      A1=PI/2.
      A2=(FRHO(1)*FRHO(2)*FRHO(3))/(FZRHO**2)
      A3=(FRHO(1)**2-FZRHO**2)*FRHO(2)*FRHO(3)/((FRHO(1)**2-FRHO(2)
     1**2)*(FRHO(3)**2-FRHO(1)**2))
      A4=(FRHO(2)**2-FZRHO**2)*FRHO(1)*FRHO(3)/((FRHO(1)**2-FRHO(2)
     1**2)*(FRHO(2)**2-FRHO(3)**2))
      A5=(FRHO(3)**2-FZRHO**2)*FRHO(1)*FRHO(2)/((FRHO(2)**2-FRHO(3)
     1**2)*(FRHO(3)**2-FRHO(1)**2))
      B2=(FJ(1)*FJ(2)*FJ(3))/(FZJ**2)
      B3=(FJ(1)**2-FZJ**2)*FJ(2)*FJ(3)/((FJ(1)**2-FJ(2)**2)*
     1(FJ(3)**2-FJ(1)**2))
      B4=(FJ(2)**2-FZJ**2)*FJ(1)*FJ(3)/((FJ(1)**2-FJ(2)**2)*
     1(FJ(2)**2-FJ(3)**2))
      B5=(FJ(3)**2-FZJ**2)*FJ(1)*FJ(2)/((FJ(2)**2-FJ(3)**2)*
     1(FJ(3)**2-FJ(1)**2))
      BNRHO=A1*A2*(A3+A4+A5)
      BNJ=A1*B2*(B3+B4+B5)
```

```
C
C     INPUT FOR KALMAN FILTER
      DO 110 I=1,2
      MUR(I)=0.0
      MUJ(I)=0.0
  110 CONTINUE
      DO 120 I=1,6
      IF(I.GT.3) GO TO 130
      GAMRR(I,1)=GAMRHO(I)
      GAMRR(I,2)=0.0
      GAMJR(I,1)=GAMJ(I)
      GAMJR(I,2)=0.0
      GO TO 120
  130 GAMRR(I,1)=0.0
      GAMRR(I,2)=GAMRHO(I-3)
      GAMJR(I,1)=0.0
      GAMJR(I,2)=GAMJ(I-3)
  120 CONTINUE
      DO 150 I=1,6
      DO 150 J=1,6
      IF(I.EQ.J) GO TO 160
      PHEERR(I,J)=0.0
      PHEEJR(I,J)=0.0
      GO TO 150
  160 IF(I.GT.3) GO TO 170
      PHEERR(I,J)=PHERHO(I)
      PHEEJR(I,J)=PHEEJ(I)
      GO TO 150
  170 PHEERR(I,J)=PHERHO(I-3)
      PHEEJR(I,J)=PHEEJ(I-3)
  150 CONTINUE
      DO 180 I=1,6
      IF(I.GT.3) GO TO 190
      CRR(1,I)=CRHO(I)
      CRR(2,I)=0.0
      CJR(1,I)=CJ(I)
      CJR(2,I)=0.0
      GO TO 180
  190 CRR(1,I)=0.0
      CRR(2,I)=CRHO(I-3)
      CJR(1,I)=0.0
      CJR(2,I)=CJ(I-3)
  180 CONTINUE
C
      WRITE(4,77)
   77 FORMAT(1H$,'DO YOU WISH TO SEE DATA VALUES (Y OR N)?')
      READ(4,78) INTQ
   78 FORMAT(1A1)
      IF(INTQ.NE.YES) RETURN
C
      WRITE(6,1000)
 1000 FORMAT(1H1,10X,43HSYMBOL ERROR RATE-DISPERSIVE JAMMED CHANNEL,
     1/,3X,31HDESIRED SIGNAL CHARACTERISTICS:)
      WRITE(6,1001) NOS,DELPHI,DELMEG,NSPB,TB
 1001 FORMAT(5X,4HNOS=,I2,64X,35H:BINARY PSK,NOS=1; BINARY FSK,NOS=2,
     1/,5X,7HDELPHI=,E13.6,50X,32H:FSK CARRIER PHASE DEVIATION,RAD,
     2/,5X,7HDELMEG=,E13.6,50X,27H:FSK TONE FREQUENCY,RAD/SEC,/,
     35X,5HNSPB=,I3,62X,29H:NUMBER OF SAMPLES PER SYMBOL,/,5X,
     443HTB=,E13.6,54X,34H:SYMBOL PERIOD,SECONDS,DIRECT PATH)
      WRITE(6,1100) DEL,NRHO,FZRHO,(FRHO(I),I=1,3),(GAMRHO(I),I=1,3),
     1(PHERHO(I),I=1,3),(CRHO(I),I=1,3),BNRHO
```

```
 1100 FORMAT(1HO,2X,35HDISPERSIVE CHANNEL CHARACTERISTICS:,/,
     15X,4HDEL=,E13.6,53X,23H:SPECULAR DELAY,SECONDS,/,5X,
     25HNRHO=,I2,63X,31H:ORDER OF DOPPLER-SPREAD FILTER,/,5X,
     36HFZRHO=,E15.6,49X,28H:DOPPLER FILTER ZERO FREQ,HZ,/,5X,
     48HFRHO(1)=,E13.6,49X,28H:DOPPLER FILTER POLE FREQ,HZ,/,5X,
     58HFRHO(2)=,E13.6,49X,28H:DOPPLER FILTER POLE FREQ,HZ,/,5X,
     68HFRHO(3)=,E13.6,49X,28H:DOPPLER FILTER POLE FREQ,HZ,/,5X,
     77HGAMRHO=,3(5X,E15.6),3X,28H:DOPPLER FILTER INPUT VECTOR,/,5X,
     87HPHERHO=,3(5X,E15.6),3X,33H:DOPPLER FILTER TRANSITION MATRIX,/,
     95X,5HCRHO=,2X,3(5X,E15.6),3X,29H:DOPPLER FILTER OUTPUT VECTOR,/,
     15X,6HBNRHO=,E15.6,49X,38H:EQUIV. NOISE BANDWIDTH,MULTI,NOISE,HZ)
      WRITE(6,1102) NOH,NH,FZH,(FH(I),I=1,3),(GAMH(I),I=1,3),(PHEEH(I),
     $I=1,3),(CH(I),I=1,3)
 1102 FORMAT(5X,4HNOH=,I2,64X,33H:0,NO DELAY SPREAD;1,DELAY SPREAD,/,5X,
     13HNH=,I2,65X,29H:ORDER OF DELAY-SPREAD FILTER,/,5X,
     24HFZH=,E15.6,51X,26H:DELAY FILTER ZERO FREQ,HZ,/,5X,
     36HFH(1)=,E13.6,51X,26H:DELAY FILTER POLE FREQ,HZ,/,5X,
     46HFH(2)=,E13.6,51X,26H:DELAY FILTER POLE FREQ,HZ,/,5X,
     56HFH(3)=,E13.6,51X,26H:DELAY FILTER POLE FREQ,HZ,/,5X,
     65HGAMH=,1X,3(5X,E15.6),4X,26H:DELAY FILTER INPUT VECTOR,/,5X,
     76HPHEEH=,3(5X,E15.6),4X,31H:DELAY FILTER TRANSITION MATRIX,/,5X,
     83HCH=,3X,3(5X,E15.6),4X,27H:DELAY FILTER OUTPUT VECTOR)
      WRITE(6,1200) NJ,DELMEJ,FZJ,(FJ(I),I=1,3),(GAMJ(I),I=1,3),
     $(PHEEJ(I),I=1,3),(CJ(I),I=1,3),BNJ
 1200 FORMAT(1HO,2X,24HJAMMING CHARACTERISTICS:,/,5X,
     13HNJ=,I2,65X,34H:ORDER OF JAMMING GENERATOR FILTER,/,5X,
     27HDELMEJ=,E13.6,50X,31H:JAMMING CARRIER OFFSET FREQ,HZ,/,5X,
     34HFZJ=,E15.6,51X,24H:JAM FILTER ZERO FREQ,HZ,/,5X,
     46HFJ(1)=,E13.6,51X,24H:JAM FILTER POLE FREQ,HZ,/,5X,
     56HFJ(2)=,E13.6,51X,24H:JAM FILTER POLE FREQ,HZ,/,5X,
     66HFJ(3)=,E13.6,51X,24H:JAM FILTER POLE FREQ,HZ,/,5X,
     75HGAMJ=,1X,3(5X,E15.6),4X,24H:JAM FILTER INPUT VECTOR,/,5X,
     86HPHEEJ=,3(5X,E15.6),4X,29H:JAM FILTER TRANSITION MATRIX,/,5X,
     93HCJ=,3X,3(5X,E15.6),4X,25H:JAM FILTER OUTPUT VECTOR,/,5X,
     14HBNJ=,E15.6,51X,34H:EQUIV. NOISE BANDWIDTH,JAMMING,HZ)
      WRITE(6,1201) ENODBR,TBR,AEST,RHODBR,(MUR(I),I=1,2),(GAMRR(I,1),I
     1=1,3),(PHEERR(I,I),I=1,3),(CRR(1,I),I=1,3)
 1201 FORMAT(1HO,2X,19HRECEIVER REFERENCE:,/,5X,
     17HENODBR=,E15.6,48X,33H:ADDITIVE WHITE ENODB ESTIMATE,DB,/,5X,
     229HDISPERSIVE CHANNEL ESTIMATES:,/,5X,
     34HTBR=,E13.6,53X,36H:SPECULAR PATH SYMBOL PERIOD,SECONDS,/,
     45X,5HAEST=,E13.6,52X,21H:DIRECT PATH STRENGTH,/,5X,
     57HRHODBR=,E13.6,50X,30H:DIFFUSE COMPONENT STRENGTH,DB,/,5X,
     64HMUR=,2(5X,E15.6),26X,28H:SPECULAR COMPONENT STRENGTH,/,5X,
     76HGAMRR=,1X,3(5X,E15.6),3X,28H:DOPPLER FILTER INPUT VECTOR,/,5X,
     87HPHEERR=,3(5X,E15.6),3X,28H:DOPPLER FILTER TRANS MATRIX,/,5X,
     94HCRR=,3X,3(5X,E15.6),3X,29H:DOPPLER FILTER OUTPUT VECTOR)
      WRITE(6,1210) SJRDBR,( MUJ(I),I=1,2),DELJR,(GAMJR(I,1),I=1,3),
     $(PHEEJR(I,I),I=1,3),(CJR(1,I),I=1,3)
 1210 FORMAT(1HO,2X,18HJAMMING ESTIMATES:,/,5X,
     17HSJRDBR=,E13.6,50X,35H:STOCHASTIC SIGNAL/JAMMING RATIO,DB,/,5X,
     24HMUJ=,2(5X,E15.6),26X,31H:DETERMINISTIC JAMMING STRENGTH,/,5X,
     36HDELJR=,E13.6,51X,31H:JAMMING CARRIER OFFSET FREQ,HZ,/,5X,
     46HGAMJR=,1X,3(5X,E15.6),3X,24H:JAM FILTER INPUT VECTOR,/,5X,
     57HPHEEJR=,3(5X,E15.6),3X,29H:JAM FILTER TRANSITION MATRIX,/,5X,
     64HCJR=,3X,3(5X,E15.6),3X,25H:JAM FILTER OUTPUT VECTOR)
      WRITE(6,1220) ENODB,RHODB,SJRDB
 1220 FORMAT(1HO,2X,20HPRIMARY DATA INPUTS:,/,5X,
     16HENODB=,E13.6,51X,37H:BIT ENERGY/NOISE SPECTRAL DENSITY,DB,/,5X,
     26HRHODB=,E13.6,51X,31H:REFLECTED/DIRECT PATH RATIO,DB,/,5X,
     36HSJRDB=,E13.6,51X,29H:DIRECT PATH/JAMMING RATIO,DB)
      WRITE(6,1230)NCASEQ
 1230 FORMAT(5X,7HNCASEQ=,I6,57X,21H:CAUCHY ESCAPE FIGURE/)
      RETURN
      END
```

```
      SUBROUTINE SIG(KS,BB,BBR)
      COMMON/SAMPLE/NSPB,TB,TBR
      COMMON/DELAY/DEL,DELPHI,DELMEG,DELMEJ,DELJR
      COMMON/SEED/IXS,JXS,IXR1,JXR1,IXR2,JXR2,IXN1,JXN1,IXN2,JXN2,
     1IXJ1,JXJ1,IXJ2,JXJ2
      COMMON /SIGCON/ B(50)
      INTEGER*4 IXS,JXS,IXR1,JXR1,IXR2,JXR2,IXN1,JXN1,IXN2,JXN2
      INTEGER*4 IXJ1,JXJ1,IXJ2,JXJ2
      KK=DEL*NSPB/TB
      IF(DEL.EQ.0.0) GO TO 20
      IF(KS.GT.1) GO TO 1
      DO 10 I=1,KK
      B(I)=0.
   10 CONTINUE
    1 CONTINUE
      KKS=KS-MOD(KS-1,KK)
      BBR=B(KKS)
      IF(MOD(KS-1,NSPB).NE.0) GO TO 2
      CALL RANC(IXS,JXS,QB)
      B(KKS)=AINT(QB+0.5)
      BB=B(KKS)
      RETURN
    2 CONTINUE
      KSA=KKS-1
      IF(KSA.LE.0) KSA=KK
      B(KKS)=B(KSA)
      BB=B(KSA)
      RETURN
   20 IF(MOD(KS-1,NSPB).NE.0) GO TO 25
      CALL RANC(IXS,JXS,QB)
      BB=AINT(QB+0.5)
      BBR=BB
      RETURN
   25 BB=BB
      BBR=BB
      RETURN
      END
```

```
      SUBROUTINE MODUL(KS,BB,BBR,XST,FR,GR)
      COMMON/SAMPLE/NSPB,TB,TBR
      COMMON/DELAY/DEL,DELPHI,DELMEG,DELMEJ,DELJR
      COMMON/OPTION/NOH,NOS
      COMMON/AMAG/AT,AR,AEST
      DIMENSION XST(2)
      C=1.-2.*BB
      CR=1.-2.*BBR
      TK=(KS-0.5)/NSPB
      TKMOD=(TK-IFIX(TK))*TB
      TR=(TK-DEL/TB-IFIX(TK-DEL/TB))*TB
      GO TO (1,2),NOS
    1 AT=1.
      AR=1.
      PHEET=DELPHI*C
      PHEER=DELPHI*CR
      GO TO 10
    2 AT=1.
      AR=1.
      PHEET=DELMEG*C*TKMOD
      PHEER=DELMEG*CR*TR
   10 XST(1)=AT*COS(PHEET)
      XST(2)=AT*SIN(PHEET)
      FR=AR*COS(PHEER)
      GR=AR*SIN(PHEER)
      RETURN
      END
```

```
      SUBROUTINE RHOFLT(KS,NRHO,NRHO2,XKRHO,RHO)
      COMMON/GDB/GRHO,GN,GJ,GRHOR,GNR,GJR
      COMMON/ZPWUP/GAMH,PHEEH,CH,GAMRHO,PHERHO,CRHO,GAMJ,PHEEJ,CJ
      DIMENSION WRHO(2),RHO(2),XKRHO(10)
      DIMENSION GAMH(5),PHEEH(5),CH(5),GAMRHO(5),PHERHO(5),CRHO(5),
     $GAMJ(5),PHEEJ(5),CJ(5)
      COMMON/SEED/IXS,JXS,IXR1,JXR1,IXR2,JXR2,IXN1,JXN1,IXN2,JXN2,
     1IXJ1,JXJ1,IXJ2,JXJ2
      INTEGER*4 IXS,JXS,IXR1,JXR1,IXR2,JXR2,IXN1,JXN1,IXN2,JXN2
      INTEGER*4 IXJ1,JXJ1,IXJ2,JXJ2
      CALL MARSA(IXR1,JXR1,WRHO(1))
      CALL MARSA(IXR2,JXR2,WRHO(2))
C
      CALL LPF(KS,NRHO,NRHO2,WRHO,GAMRHO,PHERHO,CRHO,XKRHO,RHO)
C
      DO 5 I=1,2
    5 RHO(I)=GRHO*RHO(I)
      RETURN
      END
```

```
      SUBROUTINE HFILTR(KS,NH,NH2,FR,GR,XKH,FHGH)
      COMMON/OPTION/NOH,NOS
      COMMON/ZPWUP/GAMH,PHEEH,CH,GAMRHO,PHERHO,CRHO,GAMJ,PHEEJ,CJ
      DIMENSION FRGR(2),FHGH(2),XKH(10)
      DIMENSION GAMH(5),PHEEH(5),CH(5),GAMRHO(5),PHERHO(5),CRHO(5),
     $GAMJ(5),PHEEJ(5),CJ(5)
      IF(NOH.NE.0) GO TO 1
      FHGH(1)=FR
      FHGH(2)=GR
      RETURN
    1 FRGR(1)=FR
      FRGR(2)=GR
C
      CALL LPF(KS,NH,NH2,FRGR,GAMH,PHEEH,CH,XKH,FHGH)
C
      RETURN
      END



      SUBROUTINE GEN(RHO,FHGH,XSR)
      DIMENSION SOOK(2,2),RHO(2),FHGH(2),XSR(2)
      SOOK(1,1)=FHGH(1)
      SOOK(1,2)=-FHGH(2)
      SOOK(2,1)=FHGH(2)
      SOOK(2,2)=FHGH(1)
      CALL MATVEC(SOOK,2,2,RHO,XSR,2,2)
      RETURN
      END



      SUBROUTINE NOISE(XN)
      COMMON/GDB/GRHO,GN,GJ,GRHOR,GNR,GJR
      DIMENSION XN(2)
      COMMON/SEED/IXS,JXS,IXR1,JXR1,IXR2,JXR2,IXN1,JXN1,IXN2,JXN2,
     1IXJ1,JXJ1,IXJ2,JXJ2
      INTEGER*4 IXS,JXS,IXR1,JXR1,IXR2,JXR2,IXN1,JXN1,IXN2,JXN2
      INTEGER*4 IXJ1,JXJ1,IXJ2,JXJ2
      CALL MARSA(IXN1,JXN1,XN(1))
      CALL MARSA(IXN2,JXN2,XN(2))
      DO 5 I=1,2
    5 XN(I)=GN*XN(I)
      RETURN
      END
```

```
      SUBROUTINE INT2(KS,NJ,NJ2,XKJ,XJ)
      COMMON/SAMPLE/NSPB,TB,TBR
      COMMON/DELAY/DEL,DELPHI,DELMEG,DELMEJ,DELJK
      COMMON/GDB/GRHO,GN,GJ,GRHOR,GNR,GJR
      COMMON/ZPWUP/GAMH,PHEEH,CH,GAMRHO,PHERHO,CRHO,GAMJ,PHEEJ,CJ
      DIMENSION WJ(2),XKJ(10),YJ(2),HJ(2,2),XJ(2)
      DIMENSION GAMH(5),PHEEH(5),CH(5),GAMRHO(5),PHERHO(5),CRHO(5),
     $GAMJ(5),PHEEJ(5),CJ(5)
      COMMON/SEED/IXS,JXS,IXR1,JXR1,IXR2,JXR2,IXN1,JXN1,IXN2,JXN2,
     1IXJ1,JXJ1,IXJ2,JXJ2
      INTEGER*4 IXS,JXS,IXR1,JXR1,IXR2,JXR2,IXN1,JXN1,IXN2,JXN2
      INTEGER*4 IXJ1,JXJ1,IXJ2,JXJ2
      CALL MARSA(IXJ1,JXJ1,WJ(1))
      CALL MARSA(IXJ2,JXJ2,WJ(2))
C
      CALL LPF(KS,NJ,NJ2,WJ,GAMJ,PHEEJ,CJ,XKJ,YJ)
C
      TK=(KS-0.5)*TB/NSPB
      DELTK=DELMEJ*TK
      HJ(1,1)=COS(DELTK)
      HJ(1,2)=-SIN(DELTK)
      HJ(2,1)=SIN(DELTK)
      HJ(2,2)=COS(DELTK)
      CALL MATVEC(HJ,2,2,YJ,XJ,2,2)
      DO 5 I=1,2
    5 XJ(I)=GJ*XJ(I)
      RETURN
      END
```

```
      SUBROUTINE DATA(XST,XSR,XJ,XN,Z)
      DIMENSION XST(2),XSR(2),XJ(2),XN(2),Z(2)
      DIMENSION HOINV(2,2)
      COMMON/MEAN/GAMM,PHEEM,CM,PIOK
      Z(1)=XST(1)+XSR(1)+XJ(1)+XN(1)
      Z(2)=XST(2)+XSR(2)+XJ(2)+XN(2)
C
C     PRE-WHITEING THE OUT-PUT Z(K)
C
      HOINV(1,1)=COS(PIOK)
      HOINV(1,2)=-SIN(PIOK)
      HOINV(2,1)=-HOINV(1,2)
      HOINV(2,2)=HOINV(1,1)
      Z(1)=Z(1)*COS(PIOK)-Z(2)*SIN(PIOK)
      Z(2)=Z(1)*SIN(PIOK)+Z(2)*COS(PIOK)
      RETURN
      END
```

```
      SUBROUTINE REFGEN(KS,M,FTR,GTR,FRR,GRR)
C     REFERENCE SIGNALS GENERATION FOR KALMAN FILTER
C     RELATED WITH SUBROUTINE MODUL
      COMMON/SAMPLE/NSPB,TB,TBR
      COMMON/DELAY/DEL,DELPHI,DELMEG,DELMEJ,DELJR
      COMMON/OPTION/NOH,NOS
      COMMON/AMAG/AT,AR,AEST
      TK=(KS-0.5)/NSPB
      TKRMOD=(TK-IFIX(TK))*TBR
      IF(NOS.NE.1) GO TO 1
      IF(M.EQ.0) PHEETR=DELPHI
      IF(M.EQ.1) PHEETR=-DELPHI
      GO TO 2
    1 IF(M.EQ.0) PHEETR=DELMEG*TKRMOD
      IF(M.EQ.1) PHEETR=-DELMEG*TKRMOD
    2 FTR=AEST*COS(PHEETR)
      GTR=AEST*SIN(PHEETR)
      FRR=FTR
      GRR=GTR
      RETURN
      END
```

```
      SUBROUTINE KALFLT(KS,NYRR,NYJR,FTR,GTR,FRR,GRR,Z,
     $XEST,VEST,INNOV,VINV,DET)
      COMMON/SAMPLE/NSPB,TB,TBR
      COMMON/AMAG/AT,HR,AEST
      COMMON/DELAY/DEL,DELPHI,DELMEG,DELMEJ,DELJR
      COMMON/GDB/GRHO,GN,GJ,GRHOR,GNR,GJR
      COMMON/ZPWUP/GAMH,PHEEH,CH,GAMRHO,PHERHO,CRHO,GAMJ,PHEEJ,CJ
      COMMON/ZPWKAL/GAMRR,PHEERR,CRR,GAMJR,PHEEJR,CJR,MUR,MUJ
      REAL INNOV,MUR,MUJ,MU
      DIMENSION GAMH(5),PHEEH(5),CH(5),GAMRHO(5),PHERHO(5),CRHO(5),
     $GAMJ(5),PHEEJ(5),CJ(5)
      DIMENSION PHEE(12,12),GAMMA(12,4),MU(6),VNNR(2,2)
      DIMENSION GAMRR(6,2),PHEERR(6,6),CRR(2,6),GAMJR(6,2),PHEEJR(6,6),
     $CJR(2,6),MUR(2),MUJ(2)
      DIMENSION HX1(2,6),HX2(2,6),HX(2,12),HMU(2,6),HJR(2,2),HTR(2,2)
      DIMENSION HRR(2,2)
      DIMENSION XEST(12),XPRED(12),VEST(12,12),VPRED(12,12),INNOV(2),
     $VINOV(2,2),GAIN(12,2),VP(12,12),PVPT(12,12),Z(2),GTG(12,12),
     $HXXPRE(2),HMMU(2),HH(2),VTH(12,2),HXVTH(2,2),VPHT(12,2),
     $VINV(2,2),GIN(12),GHX(12,12),AIDGHX(12,12)
C       EQUIVALENCE(PHEE,GTG,GHX),(PVPT,AIDGHX),(VTH,VPHT)
      N=NYRR+NYJR
C
C     GAMMA CONSTANT MATRIX(N X J) CONSISTING WITH GAMRR AND GAMJR
C
      DO 10 I=1,N
      DO 10 J=1,4
      GAMMA(I,J)=0.0
      IF(I.LE.NYRR.AND.J.LE.2) GAMMA(I,J)=GAMRR(I,J)
      IF(I.GT.NYRR.AND.J.GT.2) GAMMA(I,J)=GAMJR(I-NYRR,J-2)
   10 CONTINUE
C
C     PHEE CONSTANT MATRIX(N X N) CONSISTING WITH PHEERR AND PHEEJR
C
      DO 20 I=1,N
      DO 20 J=1,N
      PHEE(I,J)=0.0
      IF(I.LE.NYRR.AND.J.LE.NYRR) PHEE(I,J)=PHEERR(I,J)
      IF(I.GT.NYRR.AND.J.GT.NYRR) PHEE(I,J)=PHEEJR(I-NYRR,J-NYRR)
   20 CONTINUE
C
C     MU CONSTANT VECTOR
C
      DO 30 I=1,2
      MU(I)=MUR(I)
      MU(I+2)=MUJ(I)
      MU(I+4)=1.0
   30 CONTINUE
C
C     VNNR
C
      VNNR(1,1)=GNR**2.
      VNNR(1,2)=0.0
      VNNR(2,1)=0.0
      VNNR(2,2)=VNNR(1,1)
C
      TK=(KS-0.5)*TB/NSPB
C
```

```
      HRR(1,1)=FRR*GRHOR
      HRR(2,1)=GRR*GRHOR
      HRR(1,2)=-HRR(2,1)
      HRR(2,2)=HRR(1,1)
C
      HJR(1,1)=COS(DELJR*TK)*GJR
      HJR(2,1)=SIN(DELJR*TK)*GJR
      HJR(1,2)=-HJR(2,1)
      HJR(2,2)=HJR(1,1)
C
      HTR(1,1)=FTR
      HTR(1,2)=0.0
      HTR(2,1)=0.0
      HTR(2,2)=GTR
C
      CALL MATMUL(1,HRR,2,2,CRR,NYRR,HX1,2,2,2,6,2,6)
      CALL MATMUL(1,HJR,2,2,CJR,NYJR,HX2,2,2,2,6,2,6)
C
      DO 50 I=1,2
      DO 50 J=1,N
      IF(J.LE.NYRR) HX(I,J)=HX1(I,J)
      IF(J.GT.NYRR) HX(I,J)=HX2(I,J-NYRR)
   50 CONTINUE
C
      DO 60 I=1,2
      DO 60 J=1,2
      HMU(I,J)=HRR(I,J)
      HMU(I,J+2)=HJR(I,J)
      HMU(I,J+4)=HTR(I,J)
   60 CONTINUE
C
C     KALMAN FILTER
C
      IF(KS.NE.1) GO TO 1
C
C     INITIAL CONDITION XEST(0)=0.0 AND VEST(0,0)
C
      DO 5 I=1,N
    5 XEST(I)=0.0
      DO 6 I=1,N
      DO 6 J=1,N
      VEST(I,J)=0.0
      II=MOD(I-1,3)+1
      IF(I.GT.6) GO TO 7
C     IN UPPER LEFT QUADRANT
      VEST(I,I)=GAMRHO(II)*GAMRHO(II)/(1-PHERHO(II)*PHERHO(II))
      GO TO 6
C     IN LOWER LEFT QUADRANT
    7 VEST(I,I)=GAMJ(II)*GAMJ(II)/(1.-PHEEJ(II)*PHEEJ(II))
    6 CONTINUE
C
C     STEP 1
C
    1 CALL MATVEC(PHEE,N,N,XEST,XPRED,12,12)
C
C     STEP 2
C
      CALL MATMUL(2,VEST,N,N,PHEE,N,VP,12,12,12,12,12,12)
      CALL MATMUL(1,PHEE,N,N,VP,N,PVPT,12,12,12,12,12,12)
      CALL MATMUL(2,GAMMA,N,4,GAMMA,N,GTG,12,4,12,4,12,12)
      CALL MATAS(1,PVPT,GTG,VPRED,12,12,12,12,12,12,12,12)
C
```

```
C
C       STEP 3
C
        CALL MATVEC(HX,2,N,XPRED,HXXPRE,2,12)
        CALL MATVEC(HMU,2,6,MU,HMMU,2,6)
        CALL VECAS(1,HXXPRE,HMMU,HH,2)
        CALL VECAS(2,Z,HH,INNOV,2)
C
C       STEP 4
C
        CALL MATMUL(2,VPRED,N,N,HX,2,VTH,12,12,2,12,12,2)
        CALL MATMUL(1,HX,2,N,VTH,2,HXVTH,2,12,12,2,2,2)
        CALL MATAS(1,VNNR,HXVTH,VINOV,2,2,2,2,2,2,2,2)
C
C       STEP 5
C
        CALL MATMUL(2,VPRED,N,N,HX,2,VPHT,12,12,2,12,12,2)
        CALL MATINV(VINOV,VINV,DET)
        CALL MATMUL(1,VPHT,N,2,VINV,2,GAIN,12,2,2,2,12,2)
C
C       STEP 6
C
        CALL MATVEC(GAIN,N,2,INNOV,GIN,12,2)
        CALL VECAS(1,XPRED,GIN,XEST,12)
C
C       STEP 7
C
        CALL MATMUL(1,GAIN,N,2,HX,N,GHX,12,2,2,12,12,12)
        DO 14 I=1,N
        DO 14 J=1,N
        AIDGHX(I,J)=-GHX(I,J)
        IF(I.EQ.J) AIDGHX(I,J)=1.0-GHX(I,J)
   14 CONTINUE
        CALL MATMUL(1,AIDGHX,N,N,VPRED,N,VEST,12,12,12,12,12,12)
        RETURN
        END



        SUBROUTINE DCIDM(KS,INNOV,VINV,DET,SUM,S)
        REAL*4 INNOV
        COMMON/SAMPLE/NSPB,TB,TBR
        DIMENSION INNOV(2),VINV(2,2),VTV(2)
        CALL MATVEC(VINV,2,2,INNOV,VTV,2,2)
        ARG=INNOV(1)*VTV(1)+INNOV(2)*VTV(2)
        IF(DET.LE.0.0) WRITE(6,100) DET
  100   FORMAT(1X,'DET=',1E15.6)
        SUMA=-(ALOG(DET)+ARG)
        IF(MOD(KS-1,NSPB).EQ.0) GO TO 10
        SUM=SUM+SUMA
        IF(MOD(KS,NSPB).EQ.0) S=SUM
        RETURN
   10   SUM=SUMA
        S=0.0
        RETURN
        END
```

```
      SUBROUTINE STDCIM(KS,Z,SUM,S,FTR0,GTR0,FTR1,GTR1,AFSK0,
     1AFSK1,BFSK0,BFSK1,SFSK0,SFSK1)
C     DETECT THE SIGNAL IN STANDARD
      COMMON/SAMPLE/NSPB,TB,TBR
      COMMON/OPTION/NOH,NOS
      DIMENSION Z(2)
      GO TO (1,2), NOS
    1 IF(MOD(KS-1,NSPB).EQ.0) GO TO 10
      SUM=SUM+Z(2)
      IF(MOD(KS,NSPB).EQ.0) S=SUM
      RETURN
   10 SUM=Z(2)
      S=0.0
      RETURN
    2 IF(MOD(KS-1,NSPB).NE.0) GO TO 20
      AFSK0=0.
      AFSK1=0.
      BFSK0=0.
      BFSK1=0.
      SFSK0=0.
      SFSK1=0.
   20 AFSK0=AFSK0+Z(1)*FTR0+Z(2)*GTR0
      AFSK1=AFSK1+Z(1)*FTR1+Z(2)*GTR1
      BFSK0=BFSK0+Z(1)*GTR0-Z(2)*FTR0
      BFSK1=BFSK1+Z(1)*GTR1-Z(2)*FTR1
      IF(MOD(KS,NSPB).NE.0) RETURN
      SFSK0=AFSK0**2 + BFSK0**2
      SFSK1=AFSK1**2 + BFSK1**2
      RETURN
      END
```

```
      SUBROUTINE CERKAL(KS,NYRR,NYJR,S0,S1,XEST0,XEST1,VEST0,VEST1,
     $BB,ERR,ERRATE,DIB,SMEAN,ERMEAN,SVAR,ERRVAR,DATSNR)
      COMMON/SAMPLE/NSPB,TB,TBR
      DIMENSION XEST0(12),XEST1(12),VEST0(12,12),VEST1(12,12)
      N=NYRR+NYJR
C     SYMMETRIZE THE VEST AND THE XEST FOR DIRECTED DECISION
      DO 1 I=2,N
      II=I-1
      DO 1 J=1,II
      VEST0(I,J)=VEST0(J,I)
      VEST1(I,J)=VEST1(J,I)
    1 CONTINUE
      IF(KS.EQ.NSPB) ERR=0.0
      IF(S0.GT.S1) GO TO 10
      BBHAT=1.
      DO 5 I=1,N
      XEST0(I)=XEST1(I)
    5 CONTINUE
      GO TO 20
   10 BBHAT=0.0
      DO 15 I=1,N
      XEST1(I)=XEST0(I)
   15 CONTINUE
C
   20 IF(BB.EQ.BBHAT) ERRN=0.0
      IF(BB.NE.BBHAT) ERRN=1.
      ERR=ERR+ERRN
      IB=1+IFIX((KS-0.5)/NSPB)
      ERRATE=ERR/IB
      CALL ESTERR(IB,ERR,ERRATE,DIB,SMEAN,ERMEAN,SVAR,ERRVAR,DATSNR)
      RETURN
      END
```

```
      SUBROUTINE CERSTD(KS,S,SFSK0,SFSK1,BB,ERR,ERRATE,DIB,SMEAN,
     $ERMEAN,SVAR,ERRVAR,DATSNR)
      COMMON/SAMPLE/NSPB,TB,TBR
      COMMON/OPTION/NOH,NOS
      IF(KS.EQ.NSPB) ERR=0.
      GO TO (1,2), NOS
    1 IF(S.GE.0.) BBHAT=0.
      IF(S.LT.0.) BBHAT=1.
      GO TO 10
    2 IF(SFSK0.GT.SFSK1) BBHAT=0.
      IF(SFSK1.GT.SFSK0) BBHA =1.
   10 IF(BB.EQ.BBHAT) ERRN=0.
      IF(BB.NE.BBHAT) ERRN=1.
      ERR=ERR+ERRN
      IB=1+IFIX((KS-0.5)/NSPB)
      ERRATE=ERR/IB
      CALL ESTERR(IB,ERR,ERRATE,DIB,SMEAN,ERMEAN,SVAR,ERRVAR,DATSNR)
      RETURN
      END
```

```
      SUBROUTINE ESTERR(IB,ERR,ERRATE,DIB,SMEAN,ERMEAN,SVAR,
     $ERRVAR,DATSNR)
C     AVERAGE OVER PAST ALFA SYMBOLS WITH EXPONENTIAL-
C     MEMORY DECAY TO CALCULATE THE ERROR MEAN AND ERROR VARIANCE
      COMMON/DATASN/DSRLMT,ALFA
      COMMON/CRTN/NDS,NPRNT,NIB,NCASEQ
      COMMON/SAMPLE/NSPB,TB,TBR
      COMMON/RATIO/ENODB,RHODB,SJRDB,ENODBR,RHODBR,SJRDBR
      ALPHA=EXP(-1./ALFA)
      IF(IB.NE.1) GO TO 25
      SMEAN=ERRATE
      DIB=1.
      ERMEAN=SMEAN/DIB
      SVAR=0.0
      ERRVAR=0.0
      DATSNR=0.0
      RETURN
   25 PERMEN=ERMEAN
      SMEAN=ERRATE+ALPHA*SMEAN
      DIB=1.+ALPHA*DIB
      ERMEAN=SMEAN/DIB
      SVAR=(ERRATE-ERMEAN)**2+ALPHA*SVAR
      ERRVAR=SVAR/DIB
C
C     CONSTRAINT TO ESCAPE FROM PROGRAM
C     1  GIVEN DATA SNR < DATSNR
C     2  GIVEN NUMBER OF SYMBOL < IB
C     3  CAUSHY SEQUENCE CRITERION
C     IF THESE 3 CONDITION ARE SATISFIED, THEN STOP
C
      IF(ERRVAR.EQ.0.) RETURN
      DATSNR=ERMEAN**2/ERRVAR
      CASEQ=ABS((ERMEAN-PERMEN)/ERMEAN)
      CAVAL=10.**(-NCASEQ)
      IF(DATSNR.LT.DSRLMT) RETURN
      IF(IB.LT.NIB) RETURN
      IF(CASEQ.GT.CAVAL) RETURN
C
      WRITE(6,66) IB,ERR,ERRATE,ERMEAN,ERRVAR
   66 FORMAT(1X,3HIB=,I5,2X,4HERR=,E13.6,2X,7HERRATE=,E13.6,
     12X,8HERRMEAN=,E13.6,2X,7HERRVAR=,E13.6)
      WRITE(6,30) IB,DATSNR,CASEQ
   30 FORMAT(5X,'NUMBER OF SYMBOL IS GREATER THAN GIVEN NUMBER=',I7,
     1/,5X,'DATA SNR IS GREATER THAN GIVEN LIMIT=',E15.6,
     2/,5X,'CAUSHY CRITERION IS LESS THAN GIVEN LIMIT=',E15.6)
C
      WRITE(6,200) ENODB,RHODB,SJRDB,NSPB,IB,ERMEAN,ERRVAR,DATSNR
  200 FORMAT(1H0,2X,'PRIMARY DATA OUTPUTS:',/,5X,
     1'CASE NUMBER=',3(5X,E13.6),/,5X,
     2'NUMBER OF DATA SAMPLES=',I10,/,5X,
     3'NUMBER OF MESSAGE SYMBOLS=',I5,/,5X,
     4'MEAN ERROR RATE=',E13.6,/,5X,
     5'ERROR RATE VARIANCE=',E13.6,/,5X,
     6'DATA SNR=',E13.6)
      STOP
      END
```

```
      SUBROUTINE LPF(KS,N,N2,W,GI,PI,CI,XK,YK)
      DIMENSION GAM(10,2),W(2),GW(10),PIM(10,10),XK(10),
     $PXK(10),CM(2,10),GI(5),PI(5),CI(5),YK(2)
C
C     CALCULATE GAMMA MATRIX
C
      DO 10 I=1,N2
      IF(I.GT.N) GO TO 20
      GAM(I,1)=GI(I)
      GAM(I,2)=0.
      GO TO 10
   20 GAM(I,1)=0.
      GAM(I,2)=GI(I-N)
   10 CONTINUE
C
      CALL MATVEC(GAM,N2,2,W,GW,10,2)
C
C** PHI MATRIX  IS  DIAGONAL
C
      DO 30 I=1,N2
      DO 30 J=1,N2
      PIM(I,J)=0.0
      IF(I.EQ.J.AND.I.LE.N) PIM(I,J)=PI(I)
      IF(I.EQ.J.AND.I.GT.N) PIM(I,J)=PI(I-N)
   30 CONTINUE
C
C** INITIAL XK(0)
C
      IF(KS.NE.1) GO TO 60
      DO 70 I=1,N2
      IF(I.LE.N) XK(I)=SQRT(GI(I)**2/(1.-PI(I)**2))
      IF(I.GT.N) XK(I)=SQRT(GI(I-N)**2/(1.-PI(I-N)**2))
   70 CONTINUE
C
   60 CONTINUE
      CALL MATVEC(PIM,N2,N2,XK,PXK,10,10)
      CALL VECAS(1,PXK,GW,XK,N2)
C
C** C MATRIX OR H MATIRX
C
      DO 100 J=1,N2
      IF(J.GT.N) GO TO 200
      CM(1,J)=CI(J)
      CM(2,J)=0.
      GO TO 100
  200 CM(1,J)=0.
      CM(2,J)=CI(J-N)
  100 CONTINUE
C
      CALL MATVEC(CM,2,N2,XK,YK,2,10)
      RETURN
      END
```

```
      SUBROUTINE MARSA(IXA,JXA,V)
      INTEGER*4 IXA,JXA
      CALL RANC(IXA,JXA,X1)
      CALL RANC(IXA,JXA,X2)
      X1=(X1-0.5)*2
      X2=(X2-0.5)*2
    5 W=X1**2+X2**2
      IF(W.LE.1.) GO TO 10
      CALL RANC(IXA,JXA,X1)
      CALL RANC(IXA,JXA,X2)
      X1=(X1-0.5)*2
      X2=(X2-0.5)*2
      GO TO 5
   10 XX=X1*SQRT(-2.*ALOG(W)/W)
      V=X2*XX/X1
      RETURN
      END
```

```
      SUBROUTINE MATVEC(A,N,M,B,C,NA,MA)
      DIMENSION A(NA,MA),B(MA),C(NA)
      DO 1 I=1,N
      C(I)=0.0
      DO 1 J=1,M
    1 C(I)=C(I)+A(I,J)*B(J)
      RETURN
      END
```

```
      SUBROUTINE MATMUL(IMOT,A,N,M,B,L,C,NA,MA,NB,MB,NC,MC)
      DIMENSION A(NA,MA),B(NB,MB),C(NC,MC)
C     A,B,C ARE GENERAL MATRIX
C     IF A X B  = C, THEN IMOT IS 1
C     IF A X B' = C, THEN IMOT IS 2
      DO 1 I=1,N
      DO 1 J=1,L
      C(I,J)=0.0
      DO 1 K=1,M
      GO TO (2,3),IMOT
    2 B1=B(K,J)
      GO TO 1
    3 B1=B(J,K)
    1 C(I,J)=C(I,J)+A(I,K)*B1
      RETURN
      END
```

```
J       SUBROUTINE MATINV(A,AINV,DET)
        DIMENSION A(2,2),AINV(2,2)
        DET=A(1,1)*A(2,2)-A(1,2)*A(2,1)
        IF(DET.EQ.0.0) GO TO 1
        AINV(1,1)=A(2,2)/DET
        AINV(1,2)=-A(2,1)/DET
        AINV(2,1)=-A(1,2)/DET
        AINV(2,2)=A(1,1)/DET
        RETURN
      1 WRITE(6,100)
    100 FORMAT(1X,'ERROR IN MATINV IS SINGULAR')
        STOP
        END



        SUBROUTINE VECAS(IAOS,A,B,C,N)
        DIMENSION A(N),B(N),C(N)
C       A , B , C ARE VECTORS
C       IF A + B = C, THEN IAOS IS 1
C       IF A - B = C, THEN IAOS IS 2
        IF(IAOS.NE.1)GO TO 10
        DO 1 I=1,N
      1 C(I)=A(I)+B(I)
        RETURN
     10 DO 2 I=1,N
      2 C(I)=A(I)-B(I)
        RETURN
        END



        SUBROUTINE MATAS(IAOS,A,B,C,N,M,NA,MA,NB,MB,NC,MC)
        DIMENSION A(NA,MA),B(NB,MB),C(NC,MC)
C       IF A + B = C, THEN IAOS IS 1
C       IF A - B = C, THEN IAOS IS 2
        IF(IAOS.NE.1) GO TO 10
        DO 1 I=1,N
        DO 1 J=1,M
      1 C(I,J)=A(I,J)+B(I,J)
        RETURN
     10 DO 2 I=1,N
        DO 2 J=1,M
      2 C(I,J)=A(I,J)-B(I,J)
        RETURN
        END
```

6.3. INITIAL SIMULATION RESULTS

6.3.1. Overview

The simulation program, documented above, has been run sufficiently to demonstrate its performance. The program was debugged locally on the PDP-11/40. Production runs were made remotely on the CDC-6600 at Aeronautical Systems Division, Wright-Patterson AFB, Ohio.

The program is run in an overlay configuration on the PDP-11/40, in order to not exceed the available core memory of 16 K-words (Octal, 8-Bit). In the overlay mode, the program is considerably slowed and processes at a rate of one data sample every seven seconds. At ten samples per symbol, the local rate is 51 symbols per hour of run time. To simulate error rates of 10^{-3} requires running at least 3000 symbols. Locally, this would take 59 hours per run. Clearly this is not feasible. To speed up the local simulation will require modifications to the program like those outlined in Section 6.1.-11, plus the acquisition of more real memory (core) for the Minicomputer.

When run remotely on the CDC-6600, the program is not overlayed. It occupies 17,920 words, octal, in core. It processes samples at a rate of 4.65 per second. Thus, to run a 10^{-3} error rate simulation with 10 samples per symbol requires 6,500 seconds or 1.8 hours CPU time at a cost of $111 per CPU hour. To date, the Principal Investigator has not been successful in persuading the CDC-6600 operators to allow a run in excess of 4000 seconds. These 4000 second runs are allowed only on Tuesday, Thursday, or Saturday mornings between midnight and 4 a.m. Thus, presently, remote simulations of error rates less than 10^{-3} appears impractical on the ASD machine.

It is estimated that by procuring additional core memory for the PDP-11/40, the local execution time of the simulation routine can be improved by a factor of 5. Then simulation runs at 10^{-3} error rate could be accomplished locally in 12 hours or only 7 times slower than on the CDC-6600. Given the continuous access to the Minicomputer, this latter method of simulation is clearly the most practical.

6.3.2. Detailed Results

Reported herein are the first simulation results for the optimum binary detection algorithms for Frequency-Shift-Keying (FSK) and for Phase-Shift-Keying (PSK). By optimum is meant in the decision-directed, Integrated Detection, Estimation, and Identification (IDEI) sense.

For the presently reported cases, the symbol rate was chosen to be 2500 per second, which is the same rate as for the previously reported simulation for a quaternary hybrid modulation in multipath [29]. The PSK phase deviation was chosen at 0.785 radians, for a reason to be discussed below. The equivalent phase deviation for FSK was also taken at 0.785 radians, so as to be comparable with the PSK case. Thus, the frequency shifts, with respect to the carrier frequency, were plus and minus 1962.5 Hz., for $m = 0$ and 1, respectively.

The present results were obtained for two particular channel conditions. Either colored multiplicative noise with white additive noise were present or colored plus white additive noises were present but simultaneous colored additive and multiplicative noises were not used. For the multipath case, zero differential group delay was assumed between the direct and reflected paths. This is equivalent to an assumption of non-frequency-selective fading. The I/Q low-pass components of the multiplicative noise were obtained by passing independent scalar white noise processes through two separate un-coupled low-pass filters, each having the same transfer function. This is equivalent to an assumption that the Doppler spectrum of the unmodulated carrier displays even symmetry about the carrier frequency. The discrete-time filter algorithms were obtained by driving a continuous-time filter with a sampler and Zero-Order-Hold. The continuous-time filter has three adjustable real pole frequencies and one adjustable real zero frequency. For the present results, the pole frequencies were selected as 250 Hz., 625 Hz., and 2500 Hz. The zero frequency was selcted as 10,000 Hz., giving the filter an equivalent noise bandwidth (one-sided) of 275.7 Hz. For the present multipath case, no delay-spreading filtering was assumed.

For the colored additive noise case, the same filter structure was used as for multipath, driven by two independent scalar white noise processes. Thus the colored additive spectrum was assumed to be even-symmetric about the carrier frequency with an equivalent width at radio frequencies of 551.4 Hz.

The additive white noise consisted of two scalar white noise processes of equal variance which were independent of each other and all the other white noise driving functions.

In order to calibrate the simulation, runs were first made using PSK modulation and white noise only. Results of four runs were plotted in Figure 6.3.-1 upon the theoretical error rate curve, given by

$$P_L(e) = \tfrac{1}{2}[1 - \text{erf}\ (\sqrt{\frac{E(1 - \rho)}{2N_o}})] \qquad (6.3.-1)$$

where $\frac{E}{N_o}$ is given separately for PSK and for FSK by equations (4.3.-23) and (4.3.-29), respectively. The term, ρ, is correlation coefficient between the signal waveforms for m = 0 and m = 1. For PSK, ρ = -1 and for FSK, $\rho \overset{\sim}{=} 0$. For the calibration runs, using PSK, the phase deviation was chosen to be $\pi/2$ radians. Not only did the simulated error rates fall on the theoretical PSK curve, but the standard PSK detector and the IDEI detector made precisely the same errors, symbol for symbol. Thus, it appears that for white noise only, the IDEI algorithms operate precisely as the simpler standard algorithms.

In attempting to obtain simulation results for PSK in multiplicative noise, using 90^o phase deviation, it was observed that the error rate was 0.5 for all values of E/N_o. In retrospect, this behavior may be predicted analytically. To remedy this situation, the signal phase deviation was reduced from $\pi/2$ radians in order to produce an unmodulated carrier component in the transmitted signal. Such a component serves as a channel probe and enables the optimum detector to track the multiplicative noise. Figure 6.3.-2 shows the minimization of the error rate as a function of phase deviation.

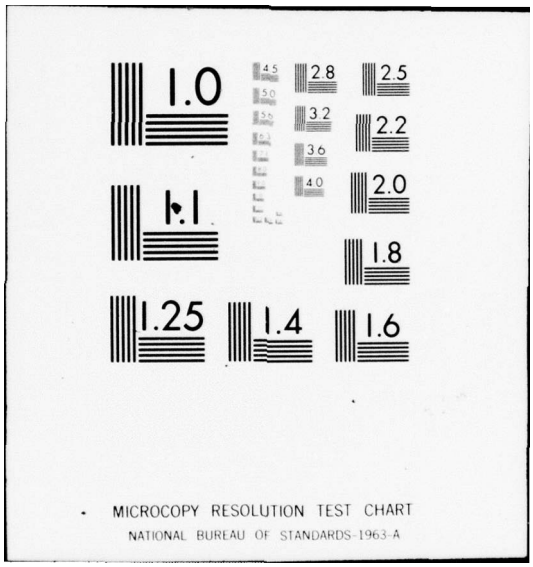
Simulation runs were made for both PSK and FSK with either multiplicative noise or additive colored noise interference. For these runs, the IDEI detection algorithms were used without using the Identification Predictor/Filter shown in Figure 4.1.-3. Rather, the Kalman filter was implemented with the exact components, $\Gamma()$, $\Phi()$, $\Lambda()$, $\underline{\mu}()$, $H_y()$, $H_\mu()$, and σ_n^2 used to generate the data as per Figure 3.6.-2. Thus, the IDEI detector was furnished with "perfect identification" of the statistics of the channel. For these

AD-A048 181 TEXAS A AND M UNIV COLLEGE STATION DEPT OF ELECTRICA--ETC F/G 17/4
LOW COST ANTI-JAM DIGITAL DATA-LINKS TECHNIQUES INVESTIGATIONS.(U)
JUN 77 J H PAINTER, J N HOLYOAK, C J YOON F33615-75-C-1011

UNCLASSIFIED AFAL-TR-77-104 NL

3 of 3
ADA048 181

END
DATE FILMED
1-78
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

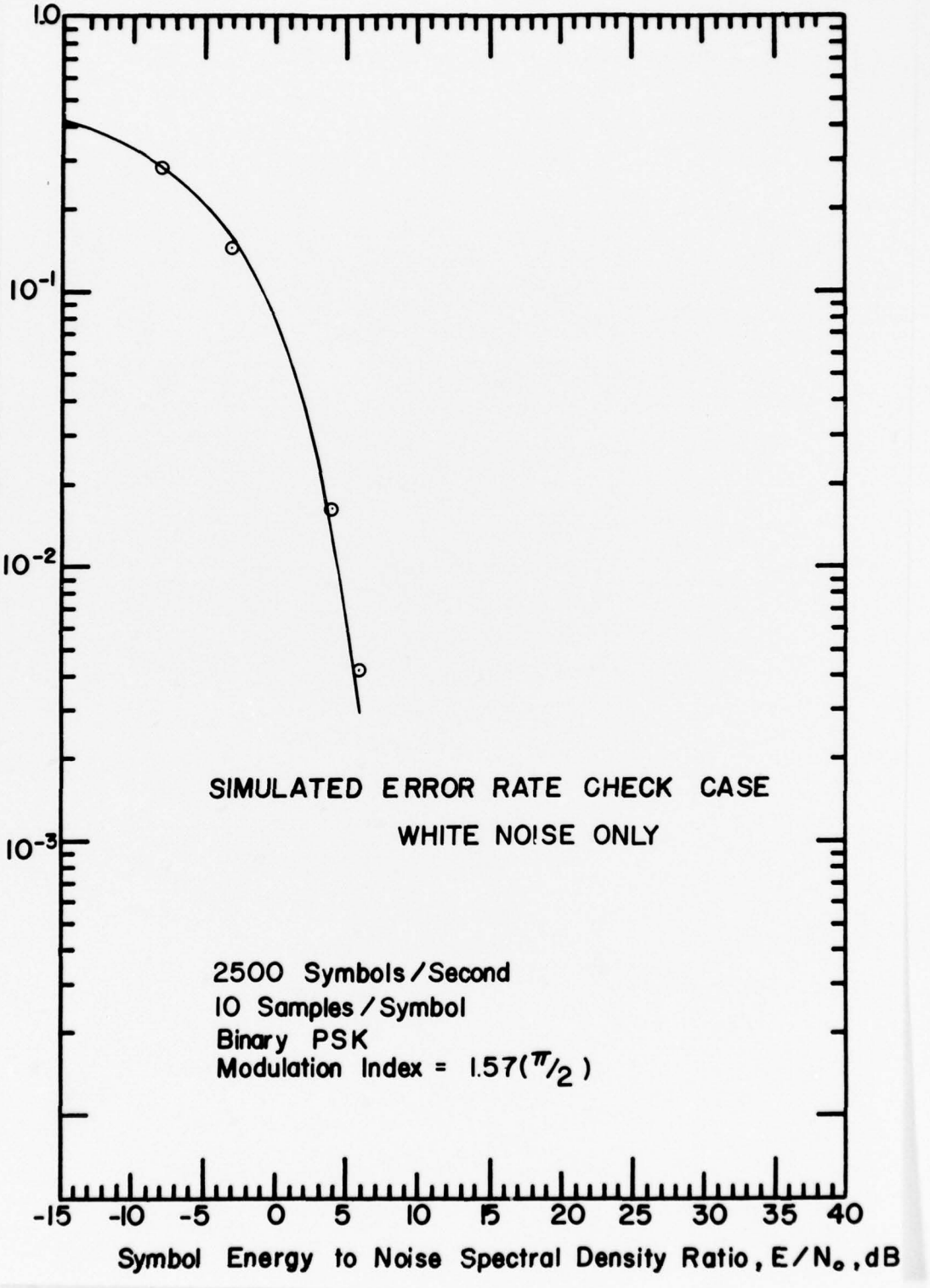


Figure 6.3.-1. Simulated Error Rate Check Case-White Noise Only

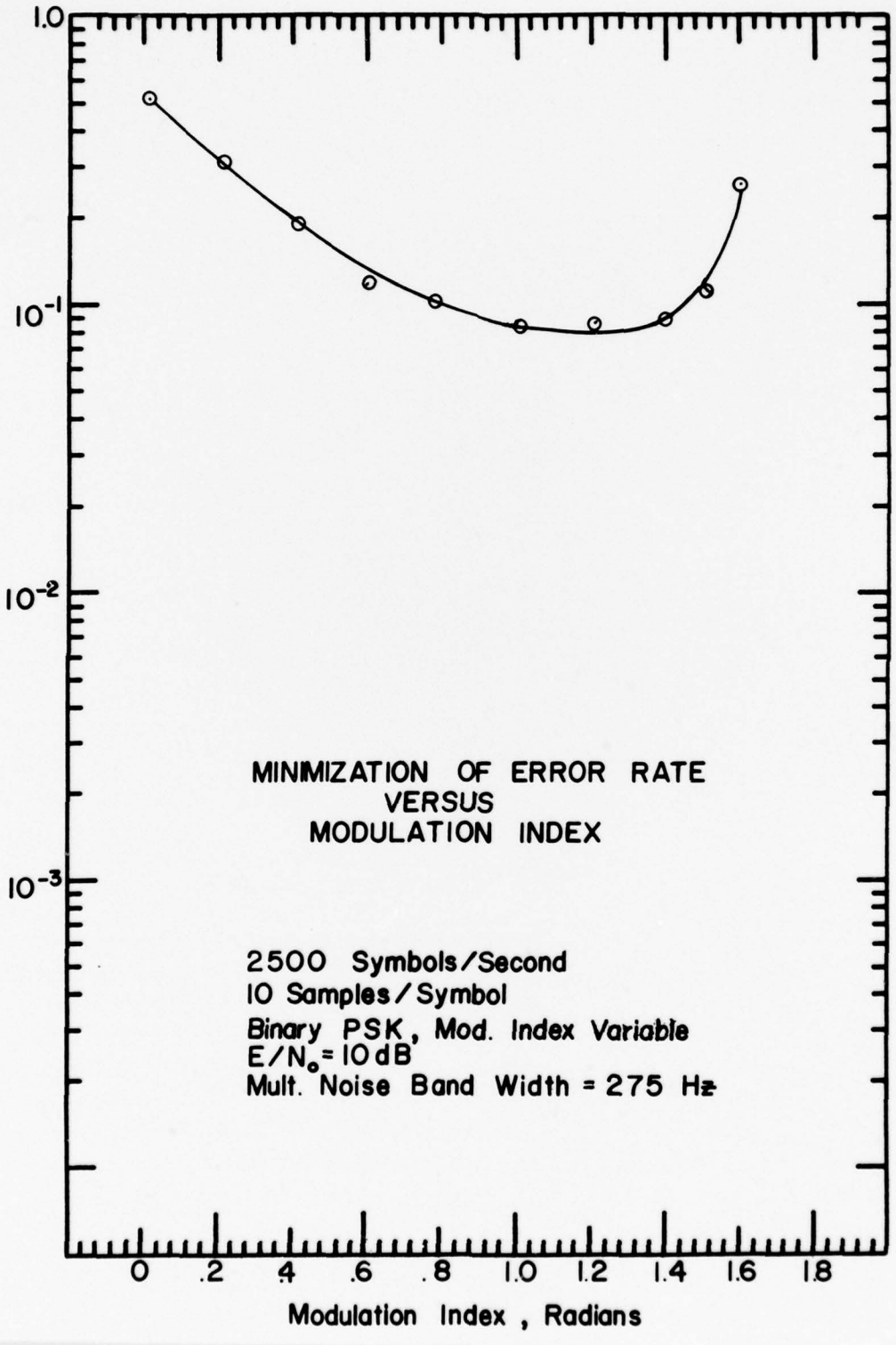


Figure 6.3.-2. Minimization of Error Rate Versus Modulation Index

runs, $H_o()$ was set to the 2 x 2 identity matrix. Thus, the data was generated without any phase perturbations in the I-Q demodulator. Perfect symbol synchronization and no quantization of the data waveforms were assumed. These runs served to determine the greatest lower bound for the error rate of the IDEI detector, without possible degradation due to imperfect identification estimators.

Results for PSK with multiplicative noise are shown in Figure 6.3.-3. The multiplicative noise level is set as though the diffuse reflected path were equal in strength to the direct path. Additive colored noise is set 53 dB below the direct path signal level. The standard detector error rate saturates at an irreducible level approximating 10^{-1}. The IDEI detector error rate decreases exponentially with increasing E/N_o. For the multiplicative noise runs it was observed that the errors occur in bursts and the smoothed error rate converges slowly. In these runs it was generally the Cauchy convergence criterion which terminated the runs.

Figure 6.3.-4 shows results for PSK with colored additive noise. The multipath is set 47 dB below the direct path signal. The colored noise is set equal to the direct path signal. The standard detector operates at an irreducible error rate level, similar to the multipath results. The IDEI detector error rate decreases exponentially with increasing E/N_o, however the penalty in E/N_o is not as large as in the multipath case. The slope of the optimum error rate curve appears to approach that of the white noise only case.

Figure 6.3.-5 shows the results for FSK with multipath of the same strength as for the PSK case. It is apparent that FSK is a more robust modulation than PSK in multiplicative moise, since both the standard detector and optimum detector perform markedly better than in the PSK case. For error rates greater than 10^{-3}, the standard detector performs almost as well as the optimum detector. Divergence in performance between the two detectors can be seen beginning for error rates less than 10^{-3}.

Figure 6.3.-6 shows the results for FSK with additive colored noise of the same strength as for the PSK case. Here again the standard detector performs considerably better than in the PSK case. However, the optimum detector comparison is much more impressive. For PSK at 10^{-3} error rate, the additive colored noise was equivalent to a 12 dB increase in the white

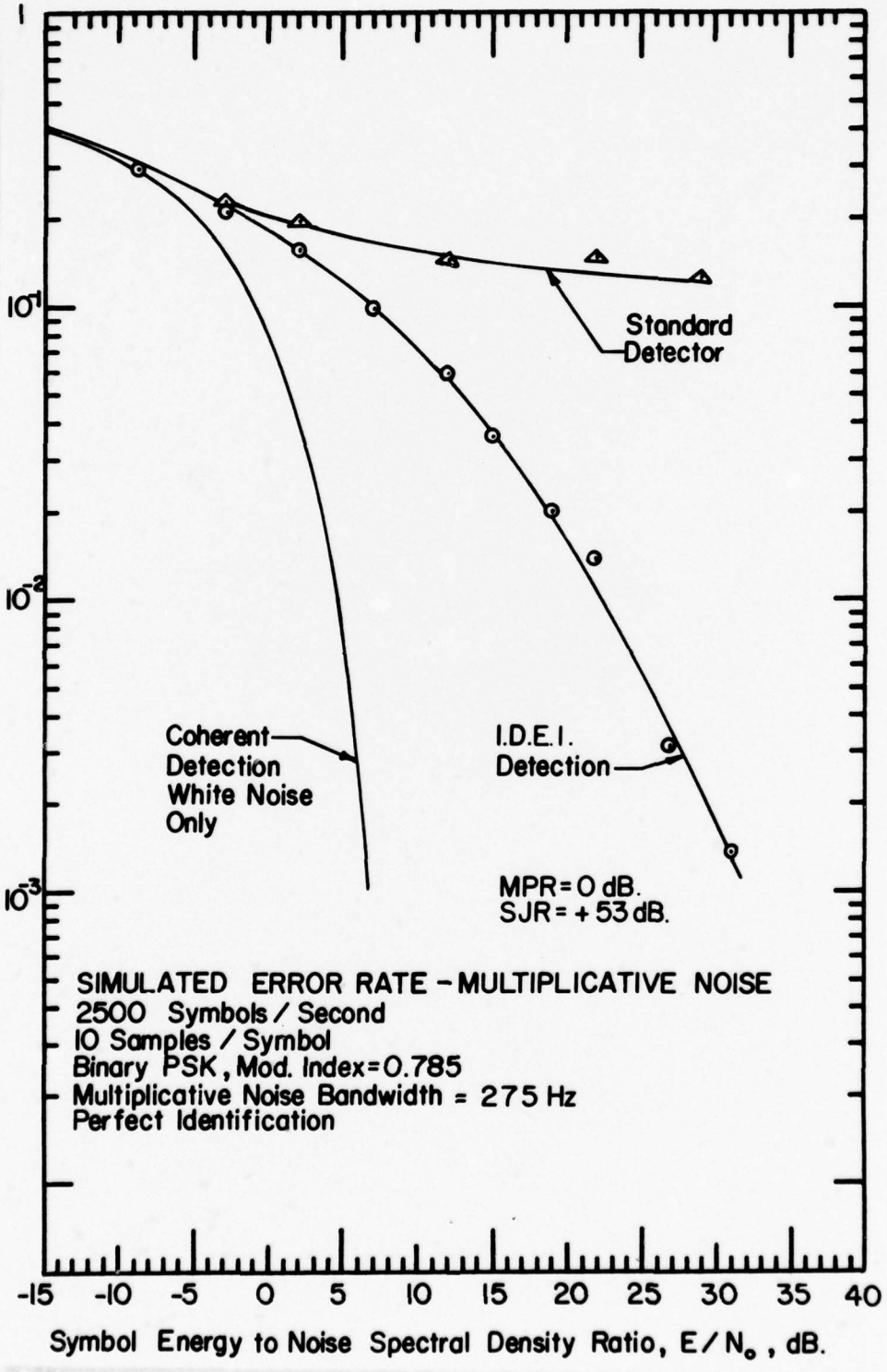


Figure 6.3.-3. Simulated Error Rate: PSK With Multipath

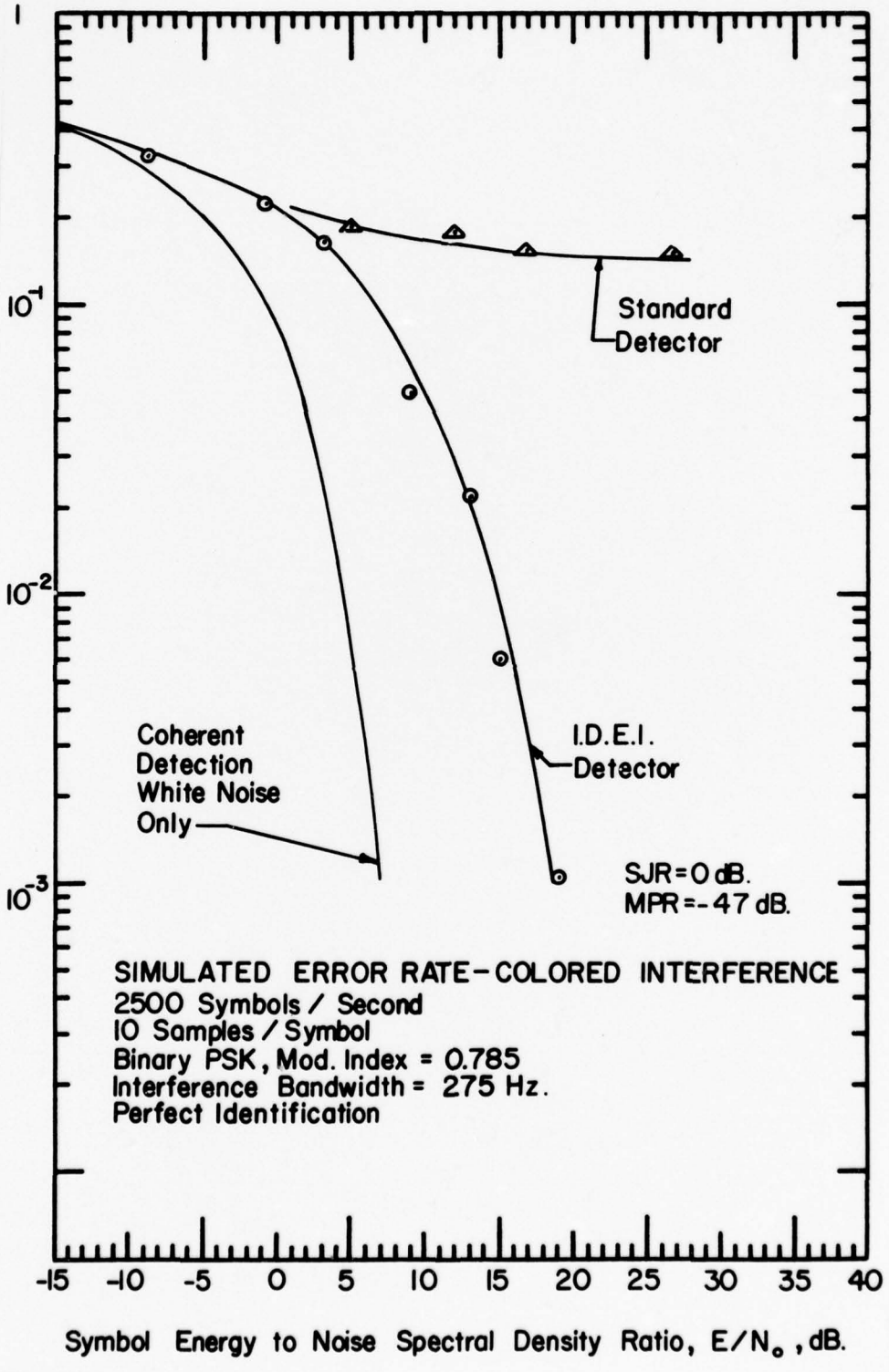


Figure 6.3.-4. Simulated Error Rate: PSK With Jamming

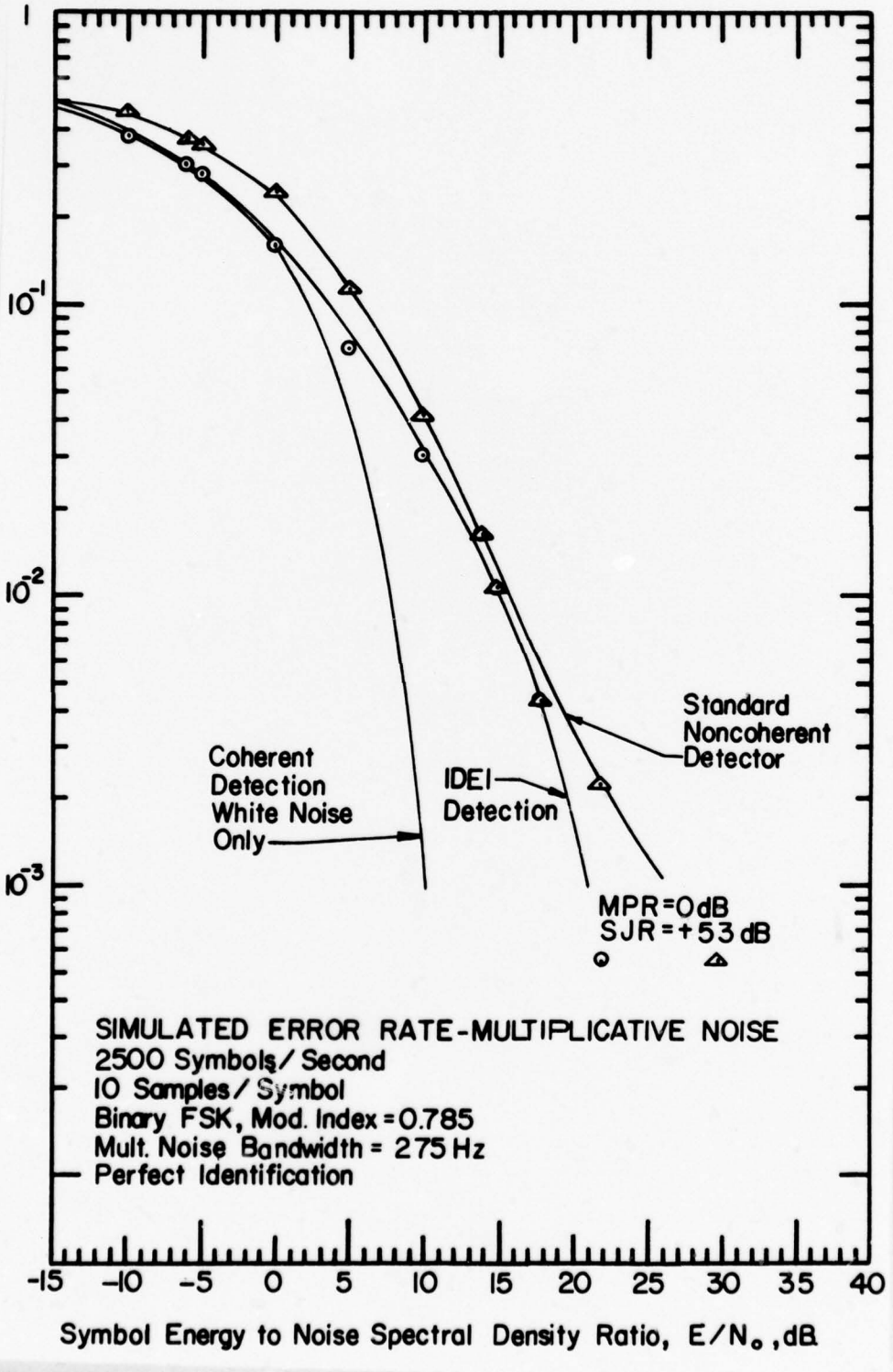


Figure 6.3.-5. Simulated Error Rate: FSK With Multipath

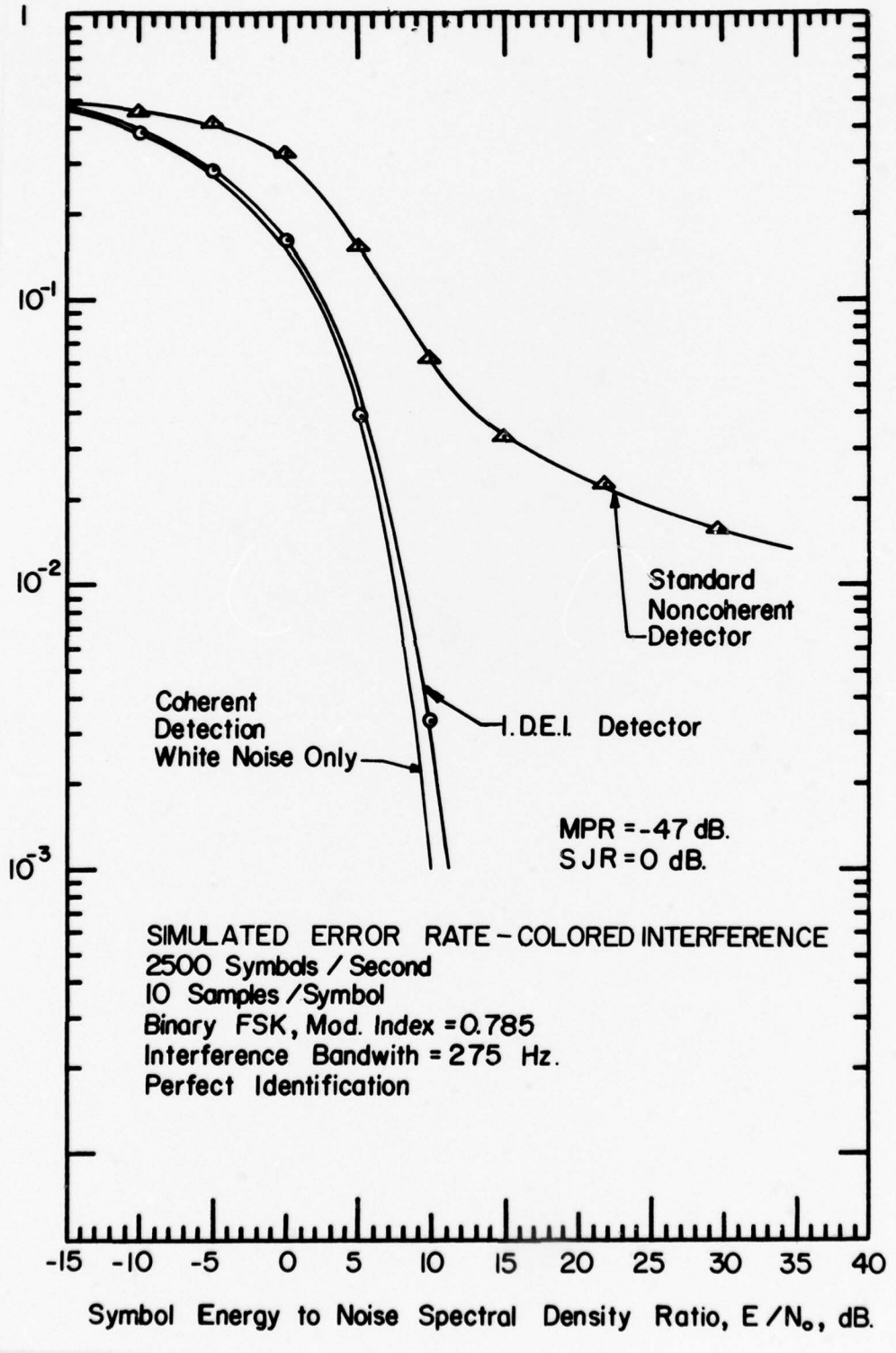


Figure 6.3.-6. Simulated Error Rate: FSK With Jamming

noise level. For FSK at 10^{-3} error rate, the same colored noise is equivalent to only a 1 dB increase in white noise level. Part of the explanation for the better performance of FSK over PSK in this simulation is that the 550 Hertz wide additive interference sits between the two FSK tones. If the colored noise were wider the FSK performance would decrease. For PSK, the interference sits in the maximum portion of the PSK signal spectrum. It should not be necessarily expected that FSK will outperform PSK when the bandwidth of the additive colored noise (jamming) is commensurate with the signal bandwidth.

SECTION VII

CONCLUSION

This report documents work done on Integrated Detection, Estimation, and Identification under the subject contract, through the final quarter of Calendar Year 1976. All of the specific tasks in the statement of work were accomplished. The work concerning Identification (See Section V) led to proposal and acceptance of a follow-on Phase II to the present contract.

It was found that the straight-forward application of recursive, sampled-data detection theory, using the Minimum Probability of Error criterion, led to signal processors which attempt to track additive jamming signals or multiplicative disturbances. The latter is a good model for signal (I-Q vector) rotations induced by either diffuse multipath or phase-detector reference jitter. The abstract, general processing algorithms appear complicated, mainly because of the discrete-time, state-variable mathematics employed. However, these general algorithms can be simplified and it is to such simplification that further effort must be applied.

Beyond simplifying the present algorithms, however, effort must now be devoted to applying the tracking idea to finding other algorithms which are perhaps slightly sub-optimal but more easily realizable in hardware. The present algorithms seem to imply that the jammer waveform must be tracked even if the jammer spectrum exceeds that of the desired signal. An immediate question is whether or not such wide-band tracking is really necessary. It would seem that, at least for finite-bandwidth signal designs such tracking might not be necessary.

The entire question of signal waveform design now seems to be reopened, due to the different performances of PSK and FSK in both jamming and multipath. Beyond waveform design lies the question of the applicability of coding to tracking-type detection. Non-bandwidth expanding codes seem particularly attractive, since the tracking technique does not require spectrum spreading to produce detection gain.

The present contract has set the stage for a detailed investigation of the IDEI technique. A flexible simulation tool now exists which can be used to probe those signalling cases for which closed form mathematical results are difficult or practically impossible to obtain. In the case of jamming

it appears (due to the efforts of the USAF Project Engineer, Mr. Mayhan) that closed-form error-rate results may be obtained. In this case, the simulation and closed-form results will be complementary. Investigation of new waveforms, closed-form results, identification, and new algorithms will be carried on in Phase II.

Finally, in order to reduce the IDEI algorithms to practice, the hardware design problem must be attacked. This problem is closely coupled with and dependent on the jammer-tracking problem. The type of hardware required will be determined only after the question of wide-band tracking is resolved.

APPENDIX A

MULTIPATH

A.1. A QUALITATIVE DESCRIPTION OF MULTIPATH

Before detailing the past work in the multipath area, it is desirable to describe the aeronautical multipath mechanism and environment. Figures A.1.-1 and A.1.-2 show the two distinct geometries involved.

In the figures, two propagation paths are shown between an aircraft and a second terminal. One is the line-of-sight, direct path. The other is a path, or set of paths, reflecting from the Earth's surface. When the surface is smooth, the waves traveling via the reflection make equal angles of incidence and reflection with the Earth's surface. The point upon the surface at which the angles are equal is called the specular reflection point, with the reflection angle, ϕ_s.

A specularly reflected signal arrives at the aircraft (or at the other terminal), attenuated in amplitude and delayed in time, but otherwise as a faithful replica of the direct path signal. The vertically polarized component may also be reversed in phase, depending on the angle of reflection. The attenuation factor is a deterministic function involving the conductivity and dielectric constant of the surface. The time delay is calculable from the geometry. The amplitudes of the reflected and direct path signals may also be affected by the differential antenna gains in the directions of the two paths.

Note that as the angular separation between the two paths becomes small at a particular terminal, it becomes increasingly difficult to suppress the reflected path signal by shaping of that terminal's antenna gain pattern. Thus, for the air-air or air-satellite cases, where the two terminals are near each other's horizon, antenna discrimination becomes practically impossible. In the air-satellite case, the angular separation between paths is small at the satellite. If the satellite is communicating with many distinct aircraft, it becomes practically impossible to use antenna discrimination at the satellite. In the air-ground case, the angular separation between paths is small at the aircraft, and a similar conclusion is reached. In this case only directivity in the ground antenna can help.

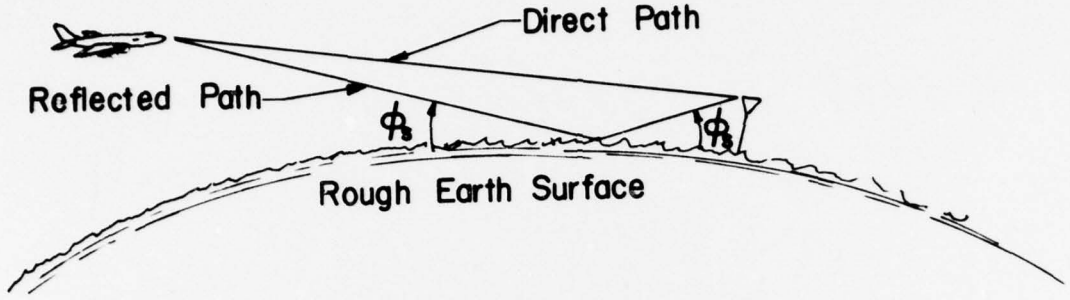


Figure A.1.-1: Geometry for Air to Ground Multipath.

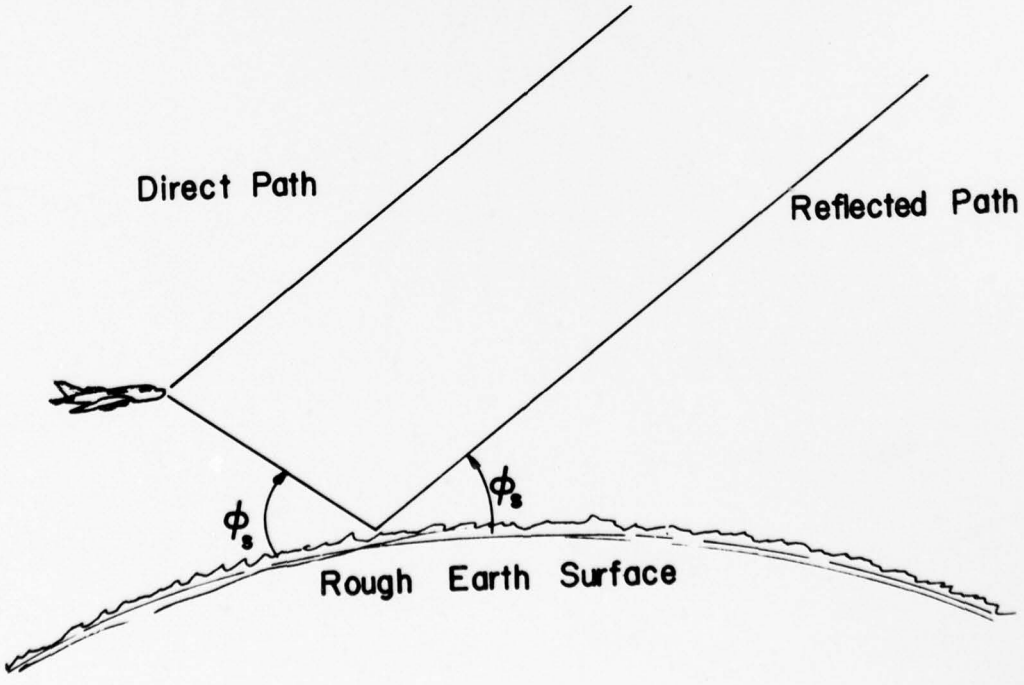


Figure A.1.-2: Geometry for Air to Satellite or Air to Air Multipath.

As the reflecting surface becomes rough, and depending on the scale of roughness with respect to wavelength, slopes may occur on the surface at other than the specular point which are of proper size and orientation to reflect waves between the two terminals. In this case, many reflected signals are received with different amplitudes, phases, and time delays. As the terminals move with respect to the Earth or to each other, the locations of the properly oriented slopes change and the amplitudes, phases and time delays of the reflected waves change with time. The terminal antennas sum these various "random" reflections. This type of reflection is called diffuse. However, for the case of slight roughness, all the reflections still appear to originate in the immediate vicinity of the specular point. However, as roughness increases, reflections occur at points farther removed from the specular point.

The qualitative effects of the two different types of reflection are, as expected, different. For specular reflection of a modulated signal whose bandwidth is not too great, the differential time delay between direct and reflected paths may not be significant in terms of the modulation waveform. That is, a bit transmitted through the reflected path may arrive at the receiver at essentially the same time as a bit propagated along the direct path. This yields what is called a non-frequency selective multipath channel. However, the differential path delay may be appreciable in terms of the phase difference between the two received carriers. As the geometry of the situation changes, the phases of the two carriers may become reversed. Under this condition the reflected path signal subtracts directly from the direct path signal, giving what is termed a specular fade.

On the other hand, a diffusely reflected signal may appear noiselike, even in the non-frequency selective case, and cannot be properly said to either add or subtract with the direct-path signal. If the transmitted signal bandwidth is not too great, then the differential time delay between all the diffusely reflecting slopes may not be significant in terms of the modulation waveform. Such a reflected signal then appears to be composed of the product of the original signal times a random low-pass noise process. This is called a Doppler-spread signal. However, if a bit reflected from one slope is received at an appreciably different time than a bit reflected at another slope (say 0.1 bit-time or greater), then the reflected signal is termed Delay-spread. The most general case is that of a Doubly-spread, frequency selective multipath channel.

For non-selective specular reflection, the only noise present is the additive receiver noise. The multipath effect is simply to reduce signal amplitude. Thus, this type of multipath may be combatted by increasing transmitted power to give a suitable margin. If the specular fades occur rapidly, the net effect is akin to a low-frequency amplitude modulation, which may or may not cause interference with signal modulation. With selective specular reflection, the reflected signal is out of synchronization with the direct-path signal and definitely acts as interference.

For non-selective, diffuse reflection, when the time variation of the reflected channel is commensurate with the time variation of the modulation, the deleterious effects on the receiver are directly proportional to the ratio of power received through the reflected path to that through the direct path. Since this ratio does not change with transmitted power level, increasing transmitted power does not increase performance. The qualitative effects are that the detection bit error rate curve, versus additive signal to noise ratio, saturates at some irreducible rate. This minimum bit error rate increases as reflected power increases.

A.2. PREVIOUS RESEARCH IN MULTIPATH COMMUNICATION.

With the availability of statistical communication theory, as made popular by Davenport and Root [61], investigators began trying to model the effects of multipath on various signals and detection schemes. In 1956, Turin [17] investigated the Maximum A Posteriori Probability continuous detection of digital signals in multipath. However, his postulated multipath channel model was very mechanistic. It consisted simply of a finite set of possible gain and delay factors which were assumed non-varying during any bit detection interval.

In 1957, Masonson [62] examined on and off keying and frequency-shift-keying under the assumption that the channel faded according to Rayleigh statistics. His channel model was also fixed during detection intervals. In 1958, Price and Green [20] reported on "A Communication Technique for Multipath Channels." This was the so-called RAKE system, which was actually built in a laboratory model. This system utilized wide-band waveforms to represent

the binary ones and zeros and employed Minimum Probability of Error detection. A great portion of the hardware was devoted to measuring the multipath channel parameters. The entire concept was based on a Turin-like model of a finite number of discrete multipaths, which did not change during the bit detection interval.

During the years from 1958 to 1966, many analyses on multipath effects were made [21-23, 63-68]. Various types of diversity were introduced and studied. Self adaptive systems were postulated. Coherent and non-coherent detections were compared. Bounds were placed on error probabilities. All of these investigations employed artificial channel models, not related to Earth-surface reflection. The most used models allowed no time-variation of the multipath disturbance during the bit detection interval. All assumed either Rayleigh (diffuse) or Rician (diffuse reflection plus direct-path) fading statistics.

In 1963, the first widely promulgated textbook treating electromagnetic wave reflection from rough Earth surfaces was published by Beckmann and Spizzichino [69]. This book made available to communication engineers a self-contained treatment of the theory and applications of rough surface propagation. Unfortunately, the reflection model was for an unmodulated carrier, non-moving terminals, and no relative movement of the reflecting surface. Also, the focus was on reflected power levels, rather than on waveforms. However, this was the first widely available work to be applied in the multipath communication area.

Applications of the Beckmann and Spizzichino models to multipath communication began to be seen after 1966. Further works using the older discrete path and Rician models continued to be seen also [70-79]. One of the interesting results in Beckmann and Spizzichino, which has apparently escaped wide notice among communication engineers, was that a reflected radio wave can never take on a Rician distribution [80]. Verifying experimental measurements were made [81]. It is possible, of course, with a totally diffuse reflection, for the direct path component to give the total received signal a Rician distribution. However, in the case of a slight to medium rough reflection, where there is a specular component, the total signal should not be expected to be Rician.

In 1968, Durrani and Staras [72] extended the Beckmann and Spizzichino model to examine, in effect, the time variation of a received unmodulated carrier at a terminal moving over a rough surface. This was done in the context of a signal from a satellite, as seen by a descending spacecraft over a planetary surface. However, this is essentially the same geometry which occurs in the air-satellite or air-air cases. It is interesting that prior to Durrani and Staras, a Russian paper had appeared in 1965, in the same area [82].

A.3. REFLECTION MODELING AND VALIDATION

A rigorous mathematical model for rough surface reflection of an arbitrarily modulated radio wave still does not exist. That is, no solution to this problem, derived from the first principles of e.m. wave propagation has ever been published. Heuristic models have been published [1, 6, 9] which appear intuitively to encompass the solution to be expected from rough surface reflection. However, these Ad Hoc models, which were originally postulated for transmission through dispersive media such as the ionosphere (or water for sonic waves), are, by their probabilistic nature, incomplete, and require the insertion of parameters to relate them explicitly to the rough surface problem. Also, the Ad Hoc models have not been completely validated by flight experiment.

It appears that the necessary parameters for a heuristic model can be inferred from the properties of rigorous models for unmodulated reflection, at least in the Doppler-spread, but non-Delay-spread case. Work to obtain rigorous unmodulated reflection models has been done by many investigators, including Beckmann and Spizzichino. It started in the early 1950's, largely under the impetus of developments in the radar field. See Barrick and Peake [83] for an excellent review.

There have been two distinct approaches to modeling rough surfaces. In the first, called the geometrical model, the surface is assumed to be composed of deterministic shapes, such as sections of cylinder, which are randomly placed on the surface. In the second, called the statistical model, the surface is assumed to be a sample function of a two-dimensional stochastic process. In the statistical model, surfaces are classified, according to the ratio of r.m.s. surface height to wavelength as slightly rough, intermediate,

or very rough. At the present time, only the slightly rough or very rough cases admit to closed form approximate mathematical solution, using the statistical model [83]. Results for the intermediate roughness case are inferred from extrapolation between the other two boundary cases. Results have been obtained for the intermediate case using the geometrical model [84]. However, due to digital computer storage limitations, these results were restricted to reflecting surfaces less than 60 wavelengths long.

For the aeronautical cases, the reflecting surfaces may be thousands of wavelengths long, since at L-band (1500 MHz) a wavelength is approximately 8 inches. Thus, the statistical treatment of rough surfaces is usually employed. Thus slightly rough and very rough cases are treated separately and results are inferred by extrapolation for the intermediate case.

For the slightly rough case, two statistical approaches have been used. The first, the so-called Perturbation model, was formulated by Rice [85] in 1951 and developed by Peake [86] in 1959, Valenzuela [87] in 1967, and Barrick and Peake [83] in 1967. The second, called the Kirchoff model, was formulated by Davies [88] in 1954, and developed in very readable form by Beckmann and Spizzichino [69]. The Perturbation model is less restrictive than the Kirchoff model. However, both give similar results for small roughness and high grazing angles. Painter et. al. [6] have applied the Kirchoff model to the problem of simulating the direct aircraft to ground station channel.

For the very rough surface, three optical models have appeared in the literature. These are called Physical Optics [69, 88-91], Ray Optics [92, 93], and Geometric Optics [94, 95]. The first Physical Optics models were scalar (polarization insensitive), using the Kirchoff approximation. Later works employed vector notations. The Ray Optics model was physically appealing, but not as mathematically rigorous as the Physical Optics model. The Geometric Optics approach applies the principal of stationary phase to evaluate the Kirchoff version of the Helmholtz integral for the complex reflected field. It provides essentially the same results as the two previous models.

Much of the impetus for deriving more and more exact models for the power reflected from rough surfaces has stemmed from the use of radar back-scatter measurements to determine the characteristic of surfaces not readily accessible to man. Examples of such surfaces are the ocean and other planets. For this reason, most of the theory cited above in the references has dealt with back-scattered and forward-scattered (bi-static) power measurements. The problem

of characterizing the spatial correlation above the surface has not received nearly so much attention, as mentioned previously [72, 73, 82]. It is interesting that the Russians apparently had the earliest documented interest in this area [96]. Another way of viewing spatial correlation is as Doppler-spreading of the back-scattered radar signal. Papers in this area are also rare [9, 97-99].

Experiments are required, both for validating precise or heuristic models, and for gathering data on which to postulate heuristic models. Several of the more recent experiments are described below.

The literature of experiments for measuring rough earth forward-scattering is sparse. The experiment by Clarke and Hendry [81], in 1963 was performed at 37.5 GHz and, thus, was more of a scaled simulation, with respect to L-band over-ocean measurement. The Russian experiment [82] in 1964 was performed using radar back-scatter. The correlation measured was that due to surface movement, rather than that due to position change above the surface. The experiment was also performed for the lower grazing angles.

A 230 MHz experiment was performed between a jet aircraft and the LES-3 satellite in 1966 [100]. Both reflected power and Doppler spreading were recorded for horizontal and vertical polarization. Both diffuse and specular multipath effects were noted. An experiment to measure specular effects was performed at L-band and S-band in 1969, using aircraft to fixed ground station transmission [101]. The reflection was from the sea at the lower grazing angles. The recorded data was matched successfully to a computer program prediction.

An interesting experiment was reported in 1970 [102]. This used data taken at UHF between aircraft and the LES-5, LES-6, and TACSAT satellites. The signals were digitally modulated. An attempt was made to fit the data fades to a two-ray smoothe earth multipath model. However, fading rates were recorded much in excess of those predicted by the model. Fading from bit-to-bit was observed. The report stated that the fading model to account for such fading was not evident. The addition of a diffuse fading mechanism to the model might have better accounted for the observed data.

Two other recent experiments are of interest here. Both were run at L-band (1550 MHz). One was for a channel between the ATS-5 satellite and a high altitude jet aircraft [103]. The other experiment was for a channel between aircraft and a high-altitude balloon, simulating a satellite [104].

The portions of the experiments which are of interest here are those measuring the diffuse reflection variance and bandwidth for an unmodulated signal. Unfortunately, the results of the balloon experiment have not been published, formally, yet. It was learned from a private communication that much difficulty was being encountered in reducing balloon data. Fortunately, the interim published data from the ATS-5 experiment appears to be of excellent quality.

An L-band (1463 MHz) Air-Ground multipath experiment was performed and reported by Painter, et. al. [6] in 1971. Results were very similar to those of Pidgeon [101]. These results have been used to verify Painter's heuristic model at low grazing angles.

In summary, the status of the mathematical modeling of reflected signals is as follows. Much theoretical and experimental work has been done for unmodulated sine waves. No rigorous model has been produced for modulated waves. Heuristic models have been postulated and some experimental validation performed. A validated heuristic model for the general multipath case is lacking.

APPENDIX B

THE GFE MINICOMPUTER FACILITY

B.1. INTRODUCTION

When the original Monte Carlo simulation was developed and used at NASA, the Principal Investigator had access to the NASA Langley CDC-6600 through a research-dedicated remote terminal. The simulation ran on the NASA machine at approximately $112 per CPU hour. At Texas A&M University, a research-dedicated remote terminal was not available connecting to the TAMU computer facility, which employs an Amdahl 470V/6 machine. Also, usage rates run considerably high than $100 per CPU hour. Thus, the Principal Investigator requested that a minicomputer remote terminal be placed at Texas A&M as Government Furnished Equipment.

The facility was procured for the USAF, through the Texas A&M Research Foundation. Due to initial funding limitations, the facility was configured first to be optimum for local use in developing the Monte Carlo routines. Sufficient hardware was procured or leased to give sufficient Remote Job Entry (RJE) capability for validating the simulation, using the CDC-6600 facility at Wright-Patterson AFB. The initial facility cost was less than $50,000. During the follow-on effort to the present contract, the facility will be up-graded sufficiently to be optimum for both local use and for RJE.

The Minicomputer Facility is structured around a basic PDP-11/40 machine. as augmented for interactive graphics, manufactured by Digital Equipment Corporation as the GT-44. Also, the system is augmented for remote job entry (RJE) communications with large host sites.

The system is used in two modes. First, local program development is accomplished, generally with the aid of graphics. Second, routines requiring large computer resources, of the CDC-6600 class, are processed remotely, using the system as an RJE port.

In order to increase the efficiency of local computation, the machine includes the hardware options, Extended Instruction Set, and Floating Point Instruction Set. The initial configuration of 16K, 16 bit words of 900 ns. core memory is being up-graded by the addition of 44K of monolithic memory. This

will enable the machine to do RJE and local usage simultaneously, or run local simulation and do local program development simultaneously. A mass storage capability of two 1.2 M-byte magnetic disks is also integral to the system. System control is via a DEC-writer with 30 character, 132-column output print format. A 300-card per minute card reader is also used for program input.

The graphics subsystem serves the important functions of editing, previewing, and plotting of program listing and output data. Graphics functions are accomplished using the 17-inch refreshed display, which is driven by a separate graphics system processor. This processor provides for character and vector generation with variable intensities, blinking modes, and light-pen sensitivity. For listing display, an 80-column format is provided.

Remote communications are available in two modes with four speed options. Synchronous communications, at 2000 baud, utilizes a leased 201-A compatible modem and dial-up telephone lines. Also available is 4800 synchronous communication, using a leased 208 modem and dial-up telephone lines. Asynchronous communications, at 300 or 1200 baud, utilize owned Vadic 3405 and GSC-108 (103-compatible) modems, respectively, and dial-up telephone lines. Present communications software includes the Amd-200-UT simulator for CDC-6600 machines and Program TERMINET for the DEC-10 machine at Air Force Avionics Laboratory.

The Minicomputer System presently contains three operating systems. These are the DOS/BATCH, RT-11, and RSX-11M operating systems. DOS is the older system, currently in Version 10, which is used strictly for support of communication software and operations. RT-11 is a real-time, single-user, operating system which supports substantial graphics capabilities. FORTRAN and BASIC Compilers are supported under RT-11, each of which are used in the graphics mode. RSX-11M is a Multi-tasking, Multi-programming operating system. Contemplated usage of RSX-11M is for simultaneous RJE communications and local graphics-oriented program development.

The configuration of the present Minicomputer Terminal has been chosen to give a cost-effective trade-off between local and remote computer processing. The combination of sophisticated graphics and relatively inexpensive communications allows an optimum balance between local program development and remote, large volume, compute-bound operations. The system is expandable into an ultimate form to include 96K words of core memory, a high-speed line printer, and an industry-compatible 9-track magnetic tape transport. Such hardware

increments would each yield an increase in the efficiency of the combined local-remote processing function.

B.2. THE BASIC SYSTEM PACKAGE

The basic system is the DEC GT-44 multi-purpose, computer-based, graphic display system. This system package was specified because its major components were required for the remote terminal and were less expensive when procured as a package. The basic package includes:

(1) PDP 11/40 CPU W/16K word Core
(2) RK11-DE Disk Drive and Control (1.2M-byte Mag. Disk Memory)
(3) RK05-AA Disk Drive (1.2 M-byte Mag. Disk Memory)
(4) BM 873-YA Disk Bootstrap Loader
(5) VT11-AA Display Processor/Monitor W/Light Pen
(6) LA36 Keyboard/Printer Console (30 Char. per second)
(7) RT11/GT Operating System (Single User)
(8) RSX-11M Operating System (Multi User)
(9) FORTRAN/GT Compiler
(10) Installation plus 4 Training Credits

The two magnetic disks (with replaceable disks) provide memory for both the operating system software and for data. The Display Processor/Monitor with 17-inch CRT and Light Pen provide the interactive graphics capability required for program development, and for data preview, verification, and editing. The Keyboard/Printer Console is used for entering data and for system monitoring. The printer accommodates standard 132 column paper and is capable of teletype-rate output at 30 characters per second. This is equivalent to 12 lines per minute printout rate, which is too slow for efficient long production runs. The Bootstrap Loader hardware facilitates start-up of the entire system. The RT11/GT Operating System with FORTRAN/GT Compiler is the standard graphics-oriented software. The RSX-11M with FORTRAN/GT Compiler allows multiprogram applications.

B.3. AUGMENTATION PACKAGE

The basic graphics terminal was initially augmented to handle the RJE function and will be further augmented during the follow-on effort to provide the following total additional hardware and software:

(11) KE11-F Floating Point Hardware
(12) KE11-E Extended Instruction Set
(13) DOS/BATCH-11 Operating System
(14) KT11-D Memory Management Hardware (access 60K)
(15) Add-on/Add-in Monolithic Memory (44K)
(16) Printronix 300 Line Printer (300 lines/minute)
(17) CR11 Card Reader (300 cards/minute)
(18) DL11-E Asynchronous Line Interface (2 each)
(19) DP11-DA Synchronous Line Interface
(20) 108 Modem and Coupler
(21) 201A Modem and Coupler
(22) VA-3405 Modem and Coupler
(23) 208 Modem and Coupler
(24) Cabinets

The Floating Point Hardware Package is to increase the system computational capability, to aid the program development function. The Extended Instruction Set increases system computational flexibility, giving the system an instruction set more comparable to those of the large remote host machines. The Disk Operating System handles the communications between the GT-44 and the remote host machines. The KT-11-D is control and management hardware for the additional memory. This additional memory is used to make the running of simultaneous programs possible.

The line printer provides 300 Line Per Minute Output Capability, and is used for documentation of results of long production runs of simulations. The CR-11 Card Reader is used for inputting programs or data to the GT-44 disk memory or directly to the remote host machine. The DL-11-E and DP-11-DA Line Interfaces and Modems are hardware items required to couple the GT-44 to the dial-up telephone line for remote use.

B.4. OPERATIONS

For program development, the terminal is used as a stand-alone minicomputer with CRT display, input via keyboard or card reader, and output to magnetic disk memory, or high speed line printer (300 LPM). For simulation production runs, the terminal communicates, via inexpensive dial-up telephone lines, with the government-owned host computer. The simulation program and input data are loaded into the host machine from the terminal's disk memory, card-reader, and/or keyboard. After processing, simulation results are transmitted over the phone line from host computer to the terminal's magnetic disk memory for storage. The stored data is then previewed on the terminal's CRT. A selected portion of the data, or all of it, is then routed to the line-printer for documentation of the simulation results. Prior to procurement of the line printer, the LA-36 keyboard printer was used in lieu.

For operations to CDC-7600 computer systems, the 11/40 minicomputer is configured to look like a CDC-200 User Terminal. Such a terminal is a standard hard wired terminal which is used for RJE to CDC systems. The 11/40 simulates the 200 User Terminal by use of a software simulator which is available from DECUS for a small charge. This software simulator is designed to run under the DOS-08 operating system. Operation under a later release of DOS (i.e. DOS-10A) requires some modification.

The hardware required for CDC-200 User Terminal simulation includes:

(1) either a DP-11 or DU-11 synchronous interface
(2) 12K memory
(3) LA-30 or LA-36 Terminal, and
(4) 201A Modem.

Additional flexibility is afforded if the configuration includes:

(1) a CR-11 card reader
(2) one or two RK-05 disks (to provide mass storage), and
(3) a 300 LPM line printer.

The use of the CDC-200 User Terminal simulator allows 2000 Baud half-duplex synchronous data communications. This proves to be quite adequate for most of

the needs. However, it is possible to communicate at 4800 Baud by the use of a 200 User simulator which is a proprietary software package written by Oregan Research Institute. This software requires an additional 4-12K of memory and the use of a 208 Modem. It will run without modification under DOS-10A.

There are two points to note about this communication:

(1) it can run at speeds up to 4800 Baud, and
(2) it can support formatted file-structured devices.

The present communication to the DEC-10 is somewhat more limited. Presently, a program called TERMINET is used to perform the data link. This program simply makes the PDP-11/40 look as if it is a 120 character/sec teletype. There is no current provision for use of file-structured devices. However, there is a mode of operation which allows the use of a VT-11/VR17 graphics system as the terminal display.

The hardware required for this communication mode includes:

(1) a DL-11E asynchronous interface
(2) 16K memory
(3) an LA-30 or LA-36 Terminal
(4) a VT-11/VR17 graphics system, and
(5) a Vadic 1200 Baud modem.

This software is designed to run under both DOS-10A and RT-11V02B. It should be noted that RT-11 is a much simpler operating system to use.

The two points noted about the 200 User Terminal System are countered in the DEC-10 communications software.

(1) the maximum transfer speed is 1200 Baud (too slow for any major data transfer), and
(2) file-structured devices, as well as card readers and line printers, are not supported.

These two facts reduce the usefulness of this software for serious RJE. However, an alternative does exist. Digital Equipment Corporation is currently bringing out an extensive communications package which is called DECNET. One of

the possible configurations of this software allows communication between an 11/40 running under the RSX-11M operating system and a DEC-10. All RJE capabilities are supported. In addition, the data rate is even better than that provided by a 4800 Baud CDC-200 simulator since the protocol used is the newly developed DDCMP. DDCMP is more efficient than the BYSYNC protocol used for CDC and IBM communications.

The use of DECNET is being investigated. At present, it looks like the best solution to the DEC-10 communications problem.

B.5. EXPERIENCES WITH THE FACILITY

The following observations are based on experience with the facility accrued since July 1975.

1. For performing detailed simulation work, the Graphics is invaluable. The flexibility and insight accruing from being able to "see" into the workings of the many simulations routines make the different between success and failure of the overall simulation task.

2. 16K words memory is just not enough. Operations are greatly hampered by having to overlay programs in and out of the magnetic disks. This not only greatly slows computations, but causes accelerated wear of the disks and leads to more failures and maintenances.

3. For any amount of printout at all, a line printer is required. Having to use only the LA-36 keyboard printer greatly slows operations. Also, it wears out the keyboard rapidly.

4. A 9-track, industry-compatible tape drive would rapidly pay for itself in terms of magnetic disk usage. Tape reels cost approximately $10. Magnetic disks cost approximately $100. A reel of tape holds the equivalent of 2½ disks worth of data. Half the disks are presently used for back-up storage. A tape drive would markedly reduce the requirement for disks in total numbers and in replacement.

5. A full maintenance contract is a must.

6. In operating an RJE Graphics Terminal, it is imperative that at least one engineer be highly conversant with both the machine and with the problems being run on the machine. The PDP 11/40, GT-44 is a magnificent machine. But, it is two levels more sophisticated than most graduate electrical engineers are

used to, in terms of routine batch-type computer usage. The hardware and maintenance support is excellent, but software maintenance is a problem. To fully utilize the machine means that at least one of the responsible engineers must "get into" and "live with" the machine and all its software. These engineers should also be well versed in the technical aspects of the problems being run on the machine. This is a "Do It Yourself" computer, in the best sense of the phrase.

REFERENCES

1. Mischa Schwartz, William R. Bennett, and Seymour Stein, Communication Systems and Techniques, McGraw-Hill Book Co., Chap. 10, 1966.

2. (U). An Introduction to Pseudo-Random Systems, Vol. I and II, TR-104, The University of Michigan Cooley Electronics Laboratory, Contract DA-36-039-SC-78283, USASRDL #S-55089-61, Sept. 1961, SECRET.

3. (U). Spread-Spectrum Communication Techniques, Vols. I and II, Sylvania Applied Research Laboratory, Contract DA-18-119-AMC-624(X), Feb., 1965, SECRET.

4. C. R. Cahn, "Spread-Spectrum Applications and State-of-the-Art Equipments," NATO AGARD L. S.-5B, Paper no. 5, Nov. 22, 1972.

5. Harry L. Van Trees, Detection, Estimation, and Modulation Theory, Part III, John Wiley and Sons, Inc., Chap. 13, 1971.

6. John H. Painter, Someshwar C. Gupta, and Lewis R. Wilson, "Multipath Modeling for Aeronautical Communciations,: IEEE Transactions on Communications, vol. COM-21, No. 5, pp. 658-662, May 1973.

7. T. Kailath, "The Innovations Approach to Detection and Estimation Theory," Proceedings of the IEEE, vol. 58, No. 5, pp. 680-695, May 1970.

8. Edward Bedrosian, "The Analytic Signal Representation of Modulated Waveforms," Proceedings of the IRE, vol. 50, No. 10, pp. 2071-2076, Oct. 1962.

9. Phillip A. Bello, "Aeronautical Channel Characterization," IEEE Transactions on Communications, vol. COM-21, pp. 548-563, May 1973.

10. R. C. Dixon, Spread Spectrum Systems, John Wiley and Sons, 1976.

11. W. H. Peake, "Satellite to Aircraft Multipath Signals Over the Ocean," Final Report 3266-2, The Ohio State University Electro-Science Laboratory, May, 1974.

12. R. C. Dixon, Spread Spectrum Techniques, IEEE Press, 1976.

13. J. S. Meditch, Stochastic Optimal Linear Estimation and Control, McGraw-Hill Book Co., p. 100, 1969.

14. Brian D. O. Anderson and John B. Moore, "The Kalman-Bucy Filter as a True Time-Varying Wiener Filter," IEEE Trans. on Systems, Man, and Cybernetics, vol. SMC-1, No. 2, pp. 119-128, April 1971.

15. Le Hung Son and B. D. O. Anderson, "Design of Kalman Filters Using Signal-Model Output Statistics," Proceedings of the IEEE, vol. 120, No. 2, pp. 312-318, Feb. 1973.

16. Michael Athan, "The Relationship of Alternate State-Space Representations in Linear Filtering Problems," IEEE Transactions on Automatic Control, vol. AC-12, No. 6, pp. 775-776, Dec. 1967.

17. G. L. Turin, "Communication through noisy random multipath channels," in IRE Conv. Rec., Pt. 4, pp. 154-166, March 1956.

18. R. Price, "Optimum detection of random signals in noise, with application to scatter-multipath communication, I," IRE Trans. Inform. Theory, vol. IT-2, pp. 125-135, Dec. 1956

19. T. Kailath, "Correlation detection of signals perturbed by a random channel," IRE Trans. Inform. Theory, vol. IT-6, pp. 361-366, June 1960.

20. R. Price and P. E. Green, "A communication technique for multipath channels," Proc. IRE, vol. 46, pp. 555-570, March 1958.

21. J. N. Pierce, "Theoretical diversity improvement in frequency-shift keying," Proc. IRE, vol. 46, pp. 903-910, May 1958.

22. J. C. Hancock and W. C. Lindsey, "Optimum performance of self-adaptive systems operating through a Rayleigh-fading medium," IEEE Trans. on Commun. Syst., vol. CS-11, pp. 443-453, Dec. 1963.

23. R. Esposito, D. Middleton, and J. A. Mullen, "Advantages of amplitude and phase adaptivity in the detection of signals subject to slow Rayleigh fading," IEEE Trans. Inform. Theory, vol. IT-11, pp. 473-482, Oct. 1965.

24. N. E. Nahi, "Optimal Recursive Estimation with Uncertain Observation," IEEE Trans. Inform. Theory, vol. IT-15, pp. 457-462, July 1969.

25. D. G. Lainiotis, "Joint Detection, Estimation, and System Identification," Information and Control, vol. 19, pp. 75-92, 1971.

26. D. Middleton and R. Esposito, "Simultaneous optimum detection and estimation of signals in noise," IEEE Trans. Inform. Theory, vol. IT-14, pp. 434-444, May 1968.

27. A. Fredriksen, D. Middleton, and D. Vandelinde, "Simultaneous Signal Detection and Estimation under Multiple Hypotheses," IEEE Trans. Inform. Theory, vol. IT-18, pp. 607-614, Sept. 1972.

28. J. H. Painter and S. C. Gupta, "Recursive ideal observer detection of known M-ary signals in multiplicative and additive Gaussian noise," IEEE Trans. on Commun., vol. COM-21, pp. 948-953, August 1973.

29. J. H. Painter and L. R. Wilson, "Simulation Results for the decision-directed MAP receiver for M-ary signals in multiplicative and additive Gaussian noise," IEEE Trans. on Commun., vol. COM-22, pp. 649-660, May 1974.

30. R. K. Mehra, "Approaches to Adaptive filtering," IEEE Trans. Auto. Control, vol. AC-17, pp. 693-698, Oct. 1972.

31. J. G. Proakis, P. R. Drouilhet, and R. Price, "Performance of Coherent detection systems using decision-directed channel measurement," IEEE Trans. Commun. Syst., vol. COM-12, pp. 54-63, March 1964.

32. D. G. Lainiotis, "Partitioning: A unifying framework for adaptive systems, I: Estimation," Proc. IEEE, vol. 64, pp. 1126-1143, Aug. 1976.

33. Thomas Kailath, "A General Likelihood-Ratio Formula for Random Signals in Gaussian Noise," IEEE Transactions on Information Theory, vol. IT-15, No. 3, pp. 350-361, May 1969.

34. N. D. Steele, Jr., "The ARINC Plan for Implementation of Data-link for the Airlines," Paper AS-298, Proceedings of the 1971 Annual Assembly Meeting, Radio Technical Commission for Aeronautics, Washington, D.C.

35. J. M. Wozencraft and R. S. Kennedy, "Modulation and Demodulation for Probabilistic Coding," IEEE Trans. on Inform. Theory, vol. IT-12, No. 3, pp. 291-297, July 1966.

36. E. R. Berlekamp, Algebraic Coding Theory, McGraw-Hill Book Co., Inc., 1968.

37. James L. Massey, "Coding and Modulation in Digital Communications," Proceedings of the 1974 Zurich Seminar on Information Theory, Paper E2, pp. 1-4.

38. David Chase, "A Combined Coding and Modulation Approach for Communication over Dispersive Channels," IEEE Trans. Comm., vol. COM-21, pp. 159-174, Mar. 1973.

39. R. R. Mosier and R. G. Clabaugh, "Kineplex, A Bandwidth-Efficient Binary Transmission System," AIEE Transactions (Part I: Communications and Electronics), vol. 76, pp. 723-728, Jan. 1958.

40. William R. Bennett and James R. Davey, Data Transmission, McGraw-Hill Book Co., pp. 49-66, 1965.

41. Andrew P. Sage and James L. Melsa, Estimation Theory With Applications to Communications and Control, McGraw-Hill Book Co., Chap. 9, 1971.

42. Robert C. K. Lee, Optimal Estimation, Identification and Control, Research Monograph #28, The M.I.T. Press, Cambridge, Mass., 1964.

43. Andrew P. Sage and James L. Melsa, System Identification, Academic Press, 1971.

44. Daniel Graupe, Identification of Systems, Van Nostrand Reinhold Co., 1972.

45. Pieter Eykhoff, System Identification, John Wiley and Sons, 1974.

46. Brian D. O. Anderson, John B. Moore, and Sonny G. Loo, "Spectral Factorization of Time-Varying Covariance Functions," IEEE Trans. on Info. Theory, vol. IT-15, No. 5, pp. 550-557, Sept. 1969.

47. J. H. Painter, "Applications of Analytic Function Theory to Analysis of Single-Sideband Angle-Modulated Systems," NASA Technical Note TN D-5446, Appendix A, Sept. 1969.

48. V. Popov, "Invariant description of linear time-invariant controllable systems," SIAM J. Contr., vol. 10, pp. 252-264, 1972.

49. H. L. Weinert and J. J. Anton, "Canonical forms for multivariable system identification," Proc. 1972 IEEE Conf. Decision Contr., New Orleans, La., 1972.

50. M. J. Denham, "Canonical Forms for the identification of multivariable linear systems," IEEE Trans. on Auto. Contr., vol. AC-19, No. 5, Dec. 1974.

51. N. K. Sinha and P. Rozsa, "Some canonical forms for linear multivariable systems," Int. J. Control, vol. 23, No. 6, pp. 865-883, 1976.

52. R. K. Mehra, "On-line Identification of linear dynamic systems with applications to Kalman filtering," IEEE Trans. on Auto. Contr., vol. AC-16, No. 1, Feb. 1971.

53. E. Tse and H. L. Weinert, "Structure determination and parameter identification for multivariable stochastic linear systems," IEEE Trans. on Auto. Contr., vol. AC-20, No. 5, Oct. 1975.

54. T. Kailath and P. Frost, "An innovation approach to least-squares estimation, part I," IEEE Trans. Auto. Contr., vol. AC-13, pp. 646-660, 1968.

55. D. Graupe, et al, "Identification of Kalman-Bucy filters from noisy measurement anays," Int. J. Systems SCI, vol. 4, No. 5, pp. 739-756, 1973.

56. R. L. Kashyap, "Maximum likelihood identification of stochastic linear systems," IEEE Trans. on Auto. Contr., vol. AC-15, No. 1, Feb. 1970.

57. I. H. Rowe, "A statistical model for the identification of multivariable stochastic systems," IFAC Sym. on Multivariable Control Systems, Oct. 1968 Düsseldorf.

58. D. Q. Mayne, "The identification of industrial processes," Proc. IEEE Sym. on Adaptive Process, pp. 531-535, 1971.

59. W. C. Martin and A. R. Stubberud, "The innovations process with applications to identifications," Control and Dynamic Systems, vol. 12, Academic Press, 1976.

60. Bernard Friedland, "Treatment of Bias in Recursive Filtering," IEEE Trans. on Auto. Ctl, vol. AC-14, No. 4, pp. 359-367, Aug. 1969.

61. Wilbur B. Davenport, Jr. and William L. Root, An Introduction to the Theory of Random Signals and Noise, McGraw-Hill Book Co., Inc., Chapter 14, 1958.

62. M. Masonson, "Binary Transmission Through Noise and Fading", IRE National Convention Record, part 2, pp. 69-82, March 18-21, 1957.

63. G. L. Turin, "Error Probabilities for Binary Symmetric Ideal Reception Through Non-selective Slow Fading and Noise", Proceedings of the IRE, vol. 46, no. 9, pp. 1603-1619, Sept. 1958.

64. J. N. Pierce and S. Stein, "Multiple Diversity with Nonindependent Fading", Proceedings of the IRE, vol. 48, no. 1, pp. 89-104, Jan. 1960.

65. William C. Lindsey, "Error Probabilities for Rician Fading Multichannel Reception of Binary and N-ary Signals", IEEE Trans. on Info. Theory, vol. IT-10, no. 4, pp. 339-350, Oct. 1964.

66. William C. Lindsey, "Error Probabilities for Coherent Receivers in Specular and Random Channels", IEEE Trans. on Info. Theory, vol. IT-11, no. 1, pp. 147-150, Jan. 1965.

67. Phillip A. Bello, "Bounds on the Error Probability of FSK and PSK Receivers Due to Non Gaussian Noise in Fading Channels", IEEE Trans. on Info. Theory, vol. It-12, no. 3, pp. 315-326, July 1966.

68. Phillip A. Bello, "Binary Error Probabilities Over Selectively Fading Channels Containing Specular Components", IEEE Trans. on Comm. Tech., vol. COM-14, no. 4, pp. 400-406, Aug. 1966.

69. Petr Beckmann and Andre Spizzichino, The Scattering of Electrmagnetic Waves From Rough Surfaces, Pergamon Press, 1963.

70. James W. Duncan, "The Effects of Ground Reflections and Scattering on an Interferometer Direction Finder", IEEE Trans. on Aero. and Elec. Systs., vol. AES-3, nol. 6, pp. 922-932, Nov. 1967.

71. Martin Nesenbergs, "Error Probability for Multipath Fading--The "Slow and Flat" Idealization", IEEE Trans. on Comm. Tech., vol. COM-15, no. 6, pp. 797-805, Dec. 1967.

72. S. H. Durrani and H. Staras, "Multipath Problems in Communications Between Low-Altitude Spacecraft and Stationary Satellites", RCA Review, vol. , no. , pp. 77-105, Mar. 1968.

73. Harold Staras, "Rough Surface Scattering on a Communication Link", Radio Science, vol. 3, no. 6, pp. 623-631, June 1968.

74. A. E. Smith and R. S. Johnson, "A Digital Simulation of a Carrier Demodulator/Tracking Phase-locked Loop in a Noisy Multipath Environment", EASCON '68 Record, pp. 206-216, 1968.

75. J. Jay Jones, "Multichannel FSK and DPSK Reception With Three-Component Multipath", IEEE Trans. on Comm. Tech., vol. COM-16, no. 6, pp. 808-821, Dec. 1968.

76. I. S. Reed and Herman Blasbalg, "Multipath Tolerant Ranging and Data Transfer Techniques for Air-to-Ground and Ground-to-Air Links", Proc. of the IEEE, vol. 58, no. 3, pp. 422-427, March 1970.

77. Leonard Schuchman, "Wide-Band Detection of FSK Transmissions in a Three-Component-Two-Path Channel", IEEE Trans. on Comm. Tech., vol. COM-18, no. 4, pp. 319-338, Aug. 1970.

78. Leonard Schuchman, "Diversity Performance of the Wide-Band FSK System in a Three Component-Two-Path Channel", IEEE Trans. on Comm. Tech., vol. COM-13, no. 5, pp. 551-562, Oct. 1970.

79. E. I. Muehldorf, "The Effect of Multipath Reflections on Spaceborne Interferometer Accuracy", IEEE Trans. on Aero. and Elec. Systs., vol. AES-7, no. 1, pp. 122-131, Jan. 1971.

80. Op. Cit., Beckmann and Spizzichino, P. 140.

81. R. H. Clarke and G. O. Hendry, "Prediction and Measurement of the Coherent and Incoherent Power Reflected From a Rough Surface", IEEE Trans. on Antennas and Propagation, vol. AP-12, no. 3, pp. 353-363, May 1964.

82. A. I. Kalmkov, I. E. Ostrovskii, A. D. Rosenberg, and I. M. Fuchs, "Influence of the State of the Sea Surface Upon the Spatial Characteristics of Scattered Radio Signals", Izvestiya VUZ Radiofizika, vol. 8, no. 6, pp. 1117-1127, 1965.

83. Donald E. Barrick and William H. Peake, "Report on Scattering From Surfaces with Different Roughness Scales: Analysis and Interpretation", Report Number BAT-197A-10-3, Battele Memorial Institute, Nov. 1st, 1967.

84. R. R. Lentz, "A Numerical Study of Electromagnetic Scattering From Ocean-Like Surfaces", Technical Report 3030-1, The Ohio State University, April 1972.

85. S. Rice, "Reflection of Electromagnetic Waves by Slightly Rough Surfaces", in The Theory of Electromagnetic Waves, edited by M. Kline, Interscience Publishers, 1963.

86. W. Peake, "Theory of Radar Return From Terrain, IRE Convention Record, vol. 7, p. 27, 1959.

87. G. Valenzuela, "Depolarization of EM Waves by Slightly Rough Surfaces", IEEE Trans. on Ant. and Prop., vol. AP-15, no. 4, p. 532, 1967.

88. H. Davies, "The Reflection of Electromagnetic Waves From a Rough Surface", Proc. of the IEEE (Great Britain), vol. 101, part 4, p. 209, 1954.

89. M. Isakovich, "The Scattering of Waves From a Statistically Rough Surface", Zhurhal Eksperimental' noi Teoreticheskoi Fiziki, vol. 23, 1952.

90. B. Semenov, "Scattering of Electromagnetic Waves From Restricted Portions of Rough Surfaces With Finite Conductivity", Radio Tekhnika i Elektronika, vol. 10, no. 1, 1965.

91. A. Stogryn, "Electromagnetic Scattering From Rough, Finitely Conducting Surfaces", Radio Science, vol. 2, no. 4, p. 415, 1967.

92. D. Muhleman, "Radar Scattering From Venus and the Moon", The Astronomical Journal, vol. 69, no. 1, p. 34, 1964.

93. T. Hagfors, "A Study of the Depolarization of Lunar Radar Echoes", Radio Science, vol. 2, no. 4, p. 445, 1967.

94. R. Kodis, "A Note on the Theory of Scattering From an Irregular Surface", IEEE Trans. on Ant. and Prop., vol. AP-14, no. 1, p. 77, 1966.

95. D. Barrick, "Rough Surface Scattering Based on the Specular Point Theory", IEEE Trans. on Ant. and Prop., vol. AP-16, no. 4, pp. 449-454, July 1968.

96. G. A. Alekseev, "Frequency and Spatial Correlation of Waves Scattered From a Rough Surface", Izvestiya VUZ. Radio Tekhnika, vol. 9, no. 1, pp. 137-140, 1966.

97. F. B. Berger, "The Nature of Doppler Velocity Measurements", IRE Trans. on Aero. Navi. Elec., vol. ANE-4, pp. 103-112, Sept. 1957.

98. R. F. Broderick and H. S. Hayre, "Doppler Return From a Random Rough Surface", IEEE Trans. on Aero. and Elec. Systs., vol. AES-5, pp. 441-449, May 1969.

99. M. S. Sohel and H. S. Hayre, "Doppler Radar Return From Two-Dimensional Random Rough Surfaces", IEEE Trans. on Geoscience Electronics, vol. GE-10, no. 1, pp. 33-47, Jan. 1972.

100. K. L. Jordan, Jr., "Measurement of Multipath Effects in a Satellite-Aircraft UHF Link", Proc. of the IEEE, vol. 55, no. 6, pp. 117-118, June, 1967.

101. V. W. Pidgeon, "A Theoretical and Experimental Study of Radar and Multipath Effects Over the Sea", Technical Memo TG 1089, Applied Physics Laboratory, Johns Hopkins University, Oct. 1969.

102. A. L. Johnson and M. A. Miller, "Three Years of Airborne Communication Testing Via Satellite Relay", Technical Report AFAL-TR-70-156, Air Force Avionics Lab., Nov. 1970.

103. ATS-5 Multipath/Ranging Experimental Program--Interim Report, Document D6-60149, The Boeing Co., Dec. 1971.

104. Test Plan for L-band/VHF ATC Communications Experiment, Part I, The Dept. of Transportation, Transportation Systems Center, April 14, 1971.

105. G. Marsaglia, "Improving the Polar Method for Generating a Pair of Normal Random Variables," Math. Note No. 271, Math. Research Lab., Boeing Scientific Labs. 1962

106. G.E.P. Box and M.E. Muller, "A Note on the Generation of Random Normal Deviates," Annals of Math. Statistics, Vol. 29, No. 2, pp. 610-611, 1958.